



## ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO DE TELECOMUNICACIÓN

Título del proyecto:

***“MAQUETA VOIP CON MEDIA GATEWAY  
CONTROLLERS”***

Alumno: Ignacio Javier Goicoechea Bozal

Tutor: Daniel Morató Osés

Pamplona, 17 Junio 2010

# ÍNDICE

<b>1. RESUMEN .....</b>	<b>4</b>
<b>2. INTRODUCCIÓN.....</b>	<b>5</b>
2.1 ARQUITECTURA SOFTSWITCH .....	5
2.2 COMPARATIVA SOLUCIONES SOFTWARE.....	8
2.3 COMPARATIVA CODECS AUDIO .....	9
<b>3. MEDIA GATEWAY CONTROL PROTOCOL .....</b>	<b>12</b>
3.1 FUNCIONAMIENTO DEL PROTOCOLO .....	12
3.1.1 <i>ENDPOINTS MGCP</i> .....	13
3.1.2 <i>CONEXIONES MGCP</i> .....	13
3.1.3 <i>COMANDOS MGCP</i> .....	14
3.1.4 <i>SEGURIDAD EN MGCP</i> .....	16
3.1.5 <i>TRAMAS DE EJEMPLO</i> .....	16
<b>4. ASTERISK .....</b>	<b>30</b>
4.1 INTRODUCCIÓN AL SOFTWARE Y A SU FUNCIONALIDAD .....	30
4.2 CONFIGURACIÓN .....	30
4.2.1 <i>SERVIDOR ASTERISK</i> .....	30
4.2.2 <i>MGCP</i> .....	31
4.2.3 <i>SIP</i> .....	32
4.2.4 <i>DAHDI</i> .....	32
4.2.5 <i>PLAN DE LLAMADAS</i> .....	33
4.3 CALL DETAIL RECORD .....	39
4.3.1 <i>CREACIÓN DE LA BASE DE DATOS</i> .....	40
4.3.2 <i>CONFIGURACIÓN DE ASTERISK PARA USAR CDR</i> .....	40
4.3.3 <i>EJEMPLO DE UN REGISTRO CDR</i> .....	41
4.4 SERVICIOS DE VALOR AÑADIDO .....	42
4.4.1 <i>INTERACTIVE VOICE RESPONSE (IVR)</i> .....	42
4.4.2 <i>VOICEMAIL</i> .....	46
4.4.3 <i>ACCESO A BASES DE DATOS</i> .....	47
4.5 PROBLEMAS DE ASTERISK Y MGCP .....	49
<b>5. MEDIA GATEWAYS .....</b>	<b>50</b>
5.1 HARDWARE EMPLEADO.....	50
5.2 CONFIGURACIÓN .....	51
<b>6. TELÉFONOS IP .....</b>	<b>56</b>
6.1 FLASH DEL FIRMWARE .....	56
6.2 CONFIGURACIÓN .....	58
6.3 PROBLEMAS .....	60

<b>7. CONFIGURACIÓN DE SOLUCIÓN COMPLEJA.....</b>	<b>61</b>
7.1 ARMARIOS SIMULANDO DISTINTAS SEDES .....	61
7.2 ENRUTAMIENTO DE LLAMADAS (IP o LÍNEA ANALÓGICA) .....	61
7.3 CALIDAD DE SERVICIO (QOS) .....	64
7.3.1 ESCENARIOS .....	65
7.3.2 ESTUDIO TEÓRICO .....	66
7.3.3 ESTUDIO PRÁCTICO EN LABORATORIO .....	72
7.3.4 LIMITACIÓN DEL ANCHO DE BANDA .....	85
7.4 CONFIGURACIÓN EN ALTA DISPONIBILIDAD .....	87
<b>8. ANEXOS .....</b>	<b>91</b>
8.1 INSTALACIÓN ASTERISK .....	91
8.2 INSTALACIÓN POSTFIX .....	93
8.3 INTERCONEXIÓN DE MGCs (SIP vs H.323) .....	97
8.4 INSTALACIÓN DAHDI .....	98
8.5 INSTALACIÓN MYSQL .....	99
8.6 ARCHIVOS DE CONFIGURACIÓN .....	100
<b>9. CONCLUSIONES.....</b>	<b>113</b>
<b>10. BIBLIOGRAFÍA.....</b>	<b>115</b>

## 1. RESUMEN

El presente Proyecto Fin de Carrera tiene como objeto estudiar y comprobar el funcionamiento de un protocolo VoIP basado en una arquitectura softswitch.

La tecnología softswitch nace con el fin de facilitar la interconexión entre la red telefónica pública conmutada y redes basadas en paquetes. Por esta razón, las grandes compañías telefónicas son las principales usuarias de este tipo de arquitectura.

Se han evaluado 3 posibles protocolos para controlar la maqueta que se desea diseñar: SGCP, MGCP y MEGACO, siendo cada uno de ellos evolución del anterior. Se ha optado por MGCP debido a limitaciones que impiden el uso de MEGACO (este protocolo no es soportado ni por Asterisk ni por los routers Cisco 1760 disponibles).

La arquitectura softswitch se basa en un *Media Gateway Controller*(MGC) y varios *Media Gateways*.

El *Media Gateway Controller* es la inteligencia del sistema y se encarga de funciones como facturación, señalización y servicios de llamada.

El *Media Gateway* convierte la voz analógica en paquetes IP. El controlador le ordena cómo realizar la llamada y los eventos de los cuales debe informarle.

En primer lugar, se debe decidir el software que se va a emplear como MGC. Existen 2 soluciones principales; Una comercial, Cisco CallManager y otra libre, Asterisk. Por sus mejores características y soporte de MGCP se debería escoger CallManager. Sin embargo, el hecho de apostar por el software libre determinó la elección de Asterisk.

Para realizar las funciones de *Media Gateways* se han utilizado routers Cisco 1760 y teléfonos IP-300 disponibles en el laboratorio. A los teléfonos IP se les actualizó el *firmware* ya que únicamente soportaban SIP. No obstante, finalmente tuvieron que ser descartados por problemas de compatibilidad con MGCP.

Una vez implementada la arquitectura básica, se añadieron distintas funcionalidades como un sistema de Interactive Voice Response, Voicemail, acceso a bases de datos y registro de CDRs para facturación y seguimiento. Se decidió integrar esta maqueta con la existente en el Laboratorio de Telemática basada en SIP, logrando que usuarios de ambos protocolos pudieran comunicarse entre sí y acceder a todos los servicios.

Con el objeto de asegurar un correcto funcionamiento de las comunicaciones, se ha instalado (en una máquina virtual) un Asterisk secundario que permite disponer de un servicio de voz con alta disponibilidad.

Para dotar de más funcionalidad a la maqueta, se ha integrado con la red PSTN por 3 vías diferentes: la maqueta SIP existente, las interfaces FXO de los routers Cisco 1760 y la tarjeta PCI integrada en el ordenador con Asterisk.

Con el fin de optimizar el servicio, se ha realizado un estudio de QoS tomando como base el requerimiento de soportar un mínimo de 30 llamadas simultáneas. El diseño desarrollado cumple las recomendaciones para sistemas VoIP en términos de retardo, jitter y pérdida de paquetes. El estudio se realizó de forma teórica, obteniendo unos pesos determinados para las colas de los switches. Con esta información se configuraron los switches para comprobar la efectividad del diseño en un entorno de test en el laboratorio, con resultado satisfactorio.

## 2. INTRODUCCIÓN

---

El objeto principal del presente Proyecto Fin de Carrera es estudiar el funcionamiento de un protocolo basado en arquitectura softswitch.

En la actualidad, la infraestructura de las comunicaciones públicas conmutadas consiste en una variedad de diferentes redes, tecnologías y sistemas, la mayoría de las cuales se basan en estructuras de conmutación de circuitos. La tecnología evoluciona hacia redes basadas en paquetes y los proveedores de servicio necesitan interconectar sus clientes sin perder la fiabilidad, conveniencia y funcionalidad de las redes telefónicas públicas conmutadas.

La tecnología softswitch nace con el fin de cubrir estas necesidades.

En la actualidad existen 3 protocolos diseñados para controlar una arquitectura softswitch: SGCP, MGCP y MEGACO. Cada uno de ellos evoluciona a partir del anterior. SGCP, el más antiguo, era muy básico y no permitía auditar los extremos para comprobar su estado. Esto era un inconveniente importante y una de las principales razones por las cuales se desarrolló MGCP. Las carencias de MGCP, sobretudo en temas de seguridad, también provocaron la creación de MEGACO. MGCP basa su seguridad en el uso de IPSEC, mientras que MEGACO permite, además, la inclusión de una cabecera de autenticación en aquellos entornos en los que IPSEC no esté habilitado.

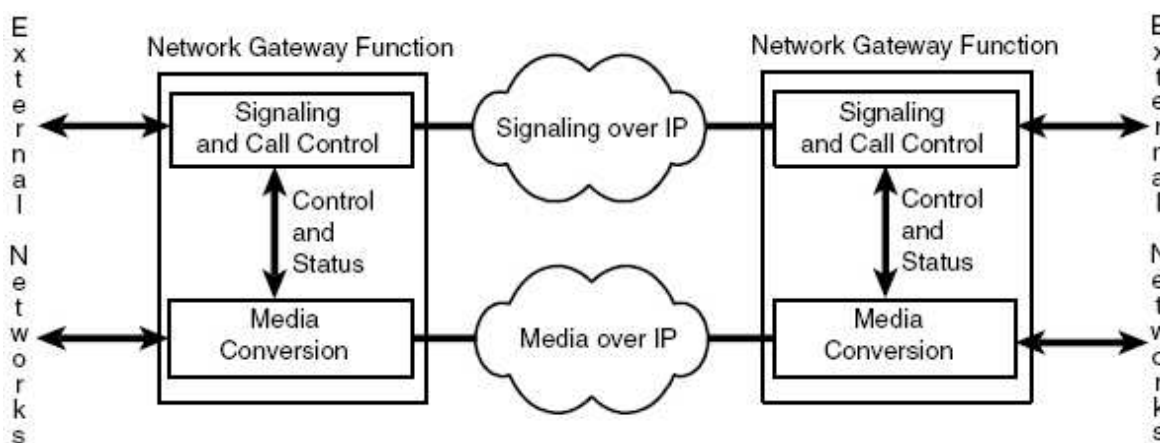
Debido a las limitaciones del material disponible que impiden el uso de MEGACO, se ha utilizado el protocolo MGCP, estudiándose un escenario con arquitectura softswitch. Las funciones de Media Gateway Controller son realizadas por el software libre Asterisk y para las funciones de Media Gateways, se dispone de routers Cisco 1760 y teléfonos IP-300. A este esquema básico se le añadieron, posteriormente, funciones típicas de una centralita PBX (llamada en espera, desvío de llamadas, conferencias...), así como servicios de valor añadido voicemail, IVR, alta disponibilidad y registro de CDRs para facturación y control de uso del sistema.

### 2.1 ARQUITECTURA SOFTSWITCH

El reto de esta arquitectura es permitir la interconexión entre telefonía analógica y VoIP. Para ello, debe cumplir dos funciones principales que, aún estando relacionadas (relación maestro-esclavo), son independientes entre sí:

1. Señalización y Control de Llamada: La señalización empleada depende de las funciones de control de llamada que estén operando en la red.
2. Conversión de datos multimedia: Se puede considerar como una función esclava ya que se encarga de cumplir los requerimientos de la señalización y control de llamada. Su principal función es satisfacer las demandas de transmisión de voz y vídeo, convirtiendo ambas señales a paquetes IP para su transporte.

La siguiente figura representa una configuración de este tipo en términos de arquitectura de red. Esta ilustración muestra un escenario donde una red VoIP se encuentra situada entre dos redes externas (analógicas), como podría ser el caso de una red VoIP que de servicio de llamadas a larga distancia. Además, ilustra la relación maestro-esclavo existente entre ambas funciones, así como la diferencia de canales de datos de cada una de ellas.



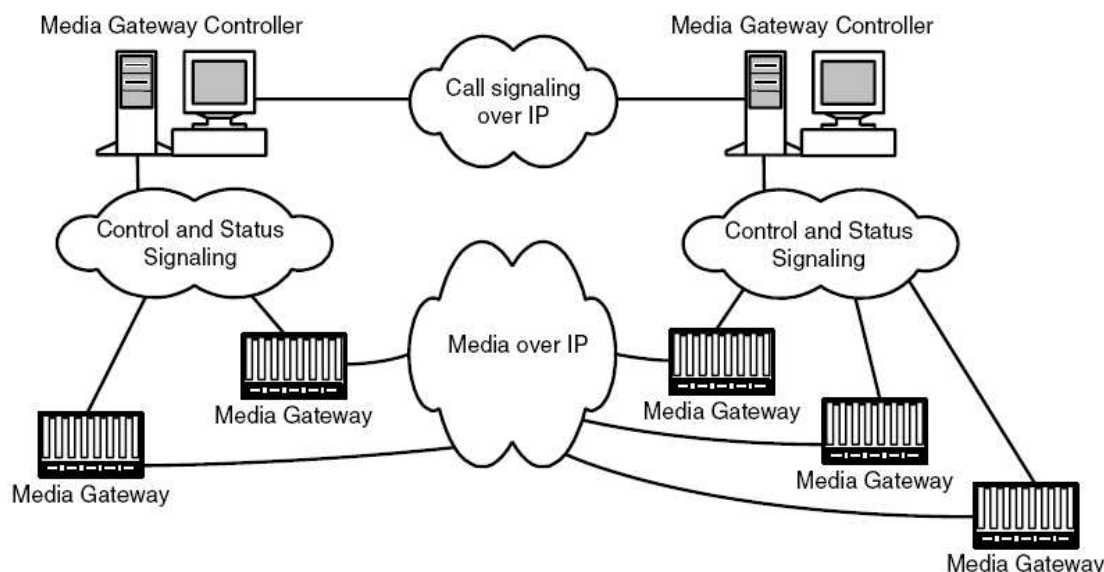
**Ilustración 2.1-1. Separación lógica de las funciones de señalización y conversión de voz y video**

La arquitectura softswitch corresponde a la **división física** de ambas funcionalidades.

Esta arquitectura es empleada habitualmente para controlar las conexiones en el punto de unión entre las redes conmutadas de paquetes y las de circuitos, descomponiéndose en un *call agent* y en un *Media Gateway*.

El *call agent* (o *Media Gateway Controller*) es la inteligencia del sistema asumiendo funciones críticas como facturación, encaminamiento de llamadas, señalización y servicios (desvíos, llamada en espera, conferencias, etc.).

El *media gateway* interconecta diferentes *streams* de voz y vídeo, creando un camino punto a punto para las llamadas. Puede poseer varias interfaces para conectarse a la red telefónica, a una red ATM, ethernet, etc. El controlador le ordena cómo realizar la conexión de la llamada con el otro extremo y los eventos de los cuales debe informarle (colgado de llamadas, códigos DTMF, etc.).



**Ilustración 2.1-2. Arquitectura softswitch**

Los componentes del softswitch generalmente residen en un edificio propiedad de la compañía telefónica llamado *central office*. Es en este lugar donde se encuentran los pares telefónicos utilizados en la telefonía analógica tradicional. Los *media gateway* se pueden conectar a diferentes dispositivos de acceso, desde un adaptador telefónico analógico (con soporte para un RJ11), hasta un *Integrated Access Device (IAD)* que soporta varios cientos de conexiones telefónicas.

Normalmente, los IADs se encuentran localizados en las centrales telefónicas cercanas a los usuarios finales, permitiendo su conexión mediante un par de hilos de cobre.

La arquitectura softswitch, ofrece una serie de ventajas:

- La conversión de los datos se realiza lo más cerca posible de la fuente.
- Los *Media Gateway Controllers* (MGCs) pueden controlar múltiples gateways situados en localizaciones diversas.
- La arquitectura es distribuida, facilitando la implantación de nuevas funcionalidades al sistema de forma rápida. Será suficiente con implementarlas en los nodos de control de llamadas (call-control nodes). Además, implica ciertas ventajas para el mundo empresarial:
  - Desde el punto de vista de un operador de red, es posible utilizar componentes de diferentes fabricantes pudiendo elegir, para cada función, el mejor equipamiento disponible en el mercado.
  - Para los fabricantes de hardware posibilita especializarse en un área concreta.

En la interconexión de redes SIP con la red telefónica pública conmutada (PSTN en inglés), el MGC gestiona la conversión de la información de señalización, mientras que los Media Gateways realizan la conversión de la voz y video.

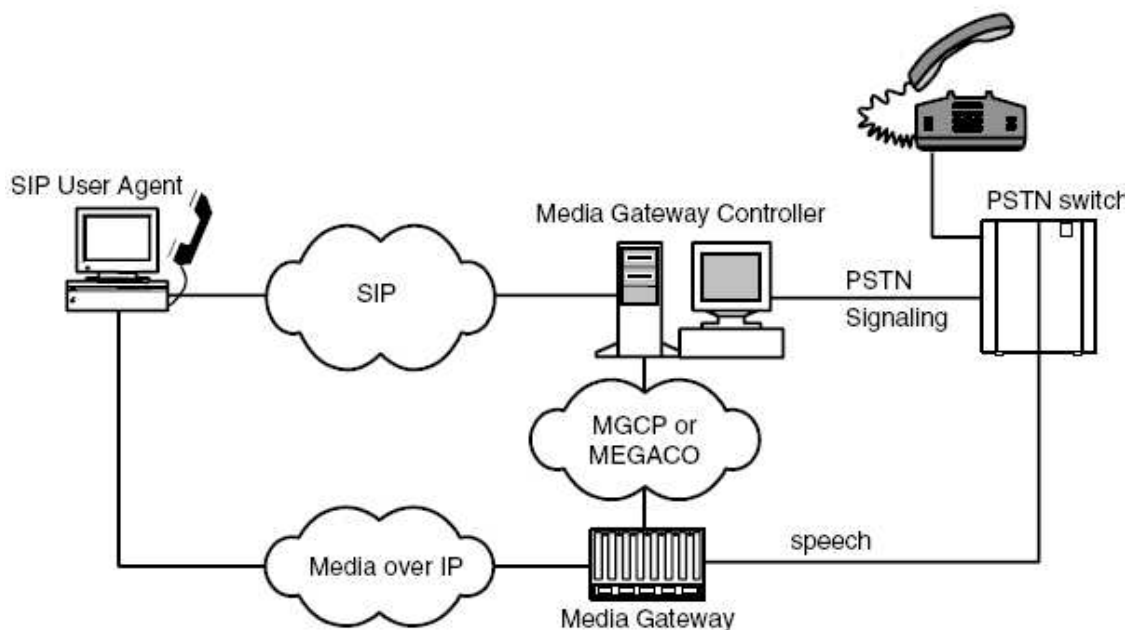


Ilustración 2.1-3. Interconexión entre SIP y PSTN usando una arquitectura softswitch

## 2.2 COMPARATIVA SOLUCIONES SOFTWARE

Para realizar las funciones de MGC existen dos firmes candidatos posibles; Asterisk basado en software libre, y Cisco CallManager como representante del software propietario. Se comparará ambas soluciones desde cinco aspectos:

1. **Coste.** Aquí Asterisk tiene una de sus principales ventajas. Los costes de equipamiento de Cisco, del software CallManager y de las licencias por cada terminal, son un claro inconveniente a la hora de desplegar (y mantener) una instalación de CallManager.
2. **Integración.** La integración de una infraestructura formada únicamente por dispositivos Cisco y gestionada por CallManager es una gran ventaja por su simplicidad y facilidad de administración. Adicionalmente, la configuración de dispositivos Cisco es más intuitiva.
3. **Sistemas Voicemail.** El voicemail gratuito incluido en Asterisk permite una mayor versatilidad que la solución de Cisco. Además, el sistema voicemail Unity empleado por Cisco implica la utilización de Ms Exchange incrementando el coste y limitando las posibilidades de elección.



4. **Teléfonos.** Para poder beneficiarse de la facilidad de configuración de CallManager, se necesita que los teléfonos asociados sean Cisco. Habitualmente, no se ofrece soporte para la instalación de dispositivos de otras marcas.
5. **Implementación de MGCP.** Asterisk tiene un soporte muy básico de MGCP no incluyendo funciones básicas como el control total de endpoints de tipo línea analógica o la solicitud de notificación de ciertos eventos.
6. **Soporte.** Cisco ofrece un servicio de soporte de mejor calidad que Asterisk aunque con un coste superior.

CallManager posee un mejor soporte para MGCP y debería ser la opción a escoger. No obstante, la apuesta por el software libre conlleva la elección de Asterisk como Media Gateway Controller.

Es deseable que dicho software soporte funcionalidades de una centralita telefónica (desvío de llamadas, voicemail, Interactive Voice Response, conferencia...). En este caso, el rango de posibles programas se ha reducido a 3: Asterisk, FreeSwitch y Objectworld Unified Communications Server.

FreeSwitch nació como un proyecto de uno de los desarrolladores de Asterisk. Se pretendía optimizar su código rescribiéndolo desde cero. Actualmente, salvo mejoras de rendimiento, no ofrece ninguna novedad respecto Asterisk y carece de algunas características que Asterisk posee. Otra interesante solución existente es Objectworld Unified Communications Server. Su principal ventaja radica en la integración con soluciones windows, gracias a la posibilidad de conexión a Directorio Activo y Ms. Exchange. Como contrapartida, destaca que es software propietario, con elevado precio de licencias e imposibilidad de añadir nuevas funcionalidades según se requieran.

Todas las soluciones PBX estudiadas poseen similares características y funcionalidades. Principalmente se diferencian en mejoras de rendimiento y/o plataformas compatibles. Por ello, puesto que se esperan resultados similares, se ha optado por la integración de los servicios PBX en Asterisk.

## 2.3 COMPARATIVA CODECS AUDIO

Imprescindible a la hora de implementar un servicio de Voz sobre IP es establecer una política de codecs de audio. En función de los codecs empleados, las prestaciones, las necesidades de ancho de banda y la capacidad del sistema variarán.

Es fundamental minimizar el *transcoding* (conversión de un códec de audio a otro códec) ya que un uso excesivo del mismo provocará un cuello de botella en el servidor Asterisk debido a la carga adicional de la transformación.. Este proceso, irremediamente, añade retardo a la comunicación y consume ciclos de procesador en el servidor. Además, los requerimientos de ancho de banda del equipo se incrementarán al no poder enviarse el audio punto a punto

FORMATO	CREADOR	COSTE	SAMPLE RATE	BIT RATE	LATENCIA	CBR	VBR
G.711	ITU-T	Gratuito	8 kHz	64kbit/s	≈ μs	Sí	No
G.722	ITU-T	Gratuito	16 kHz	64kbits/s	4ms	Sí	No
G.723.1	ITU-T	Pago	8 kHz	5.3,6.3 kbit/s	37.5ms	Sí	No
G.726	ITU-T	Gratuito	8 kHz	16,24,32,40 kbit/s	≈ μs	Sí	No
G.729	ITU-T	Pago	8 kHz	8 kbit/s	15ms	Sí	No
GSM	ETSI	Gratuito	8 kHz	13 kbit/s	20-30ms	Sí	No
iLBC	Global IP Solutions	Gratuito	8 kHz	13.33,15.20 kbit/s	20-30ms	Sí	No
Speex	Xiph.Org Foundation	Gratuito	8, 16, 32, 48 kHz	2.15-24.6 kbit/s (NB <sup>1</sup> ) 4-44.2 kbit/s (WB <sup>2</sup> )	30ms (NB) 34ms (WB)	Sí	Sí

Tabla 2.3-1 Comparativa de Codecs de Audio soportados por Asterisk

Hay ocasiones en las que evitar el *transcoding* no es posible. Conviene estudiar qué retardo introduce la conversión de un codec a otro. Cabe destacar el hecho de que algunos codecs (G.723, G.729, iLBC), Asterisk los puede utilizar pero no es capaz de realizar *transcoding* con ellos, a menos que se adquiriera la licencia correspondiente. Su uso sin *transcoding* puede ser interesante; Por ejemplo, si se debe almacenar la conversación por motivos legales y los extremos utilizan uno de los codecs mencionados.

CODEC ORIGEN	CODEC DESTINO								
	G.711u	G.711a	G.722	G.723.1	G.726	G.729	GSM	iLBC	Speex
G.711u	-	2	1001	-	1001	-	1001	-	9000
G.711a	2	-	1001	-	1001	-	1001	-	9000
G.722	1001	1001	-	-	2000	-	2000	-	9999
G.723.1	-	-	-	-	-	-	-	-	-
G.726	1001	1001	2000	-	-	-	2000	-	9999
G.729	-	-	-	-	-	-	-	-	-
GSM	2	2	1001	-	1001	-	-	-	9000
iLBC	-	-	-	-	-	-	-	-	-
Speex	1001	1001	2000	-	2000	-	2000	-	-

Tabla 2.3-2. Tiempo (en microsegundos) necesario para realizar *transcoding* en el pc del laboratorio

<sup>1</sup> NB: Narrowband. En telefonía narrowband se considera para frecuencias entre 300–3400 Hz.

<sup>2</sup> WB: Wideband. En telefonía, frecuencias desde 30 Hz. a 7000Hz.

Por su parte, los códecs que soporta el DSP de los routers 1760 son los siguientes:

- G.711a-law, G.711u-law
- G.723 Annex A(r5.3), G.723 Annex A(r6.3), G.723(r5.3), G.723(r6.3)
- G.726(r16), G.726(r24), G.726(r32)
- G.728
- G.729(br8), G.729(r8)

Con el fin de evitar incompatibilidades se ha optado por utilizar únicamente los codecs soportados por el router 1760 y que puedan ser converidos por Asterisk. Ambos condicionantes restringen los codecs a G. 711 a-law, G.711u-law y G.726.

### 3. MEDIA GATEWAY CONTROL PROTOCOL

Los requerimientos que debe cumplir el protocolo de control son los siguientes:

- Permitir la creación, modificación y eliminación de los flujos de datos en un Media Gateway.
- Capacidad de negociar los codecs y el formato a ser utilizados para la comunicación.
- Permitir al Media Gateway Controller solicitar y al MG informar sobre ciertos eventos ocurridos durante la comunicación (tonos DTMF, colgado de un extremo, etc.). El MGC debe poseer también la capacidad de especificar qué acciones tiene que ejecutar el MG al producirse dichos eventos.
- Permitir al MGC ordenar al MG el envío de señales al usuario ante ciertos eventos (Aplicar un tono de llamada cuando se descuelgue el teléfono).
- Establecer políticas de Calidad de Servicio (QoS) y permitir su modificación durante el transcurso de una llamada.
- Habilitar el reporte de estadísticas concernientes a la Calidad de Servicio. El Media Gateway Controller es ajeno a la comunicación y, por ello, los MG deben informarle de parámetros como el retardo, el jitter y la pérdida de paquetes.
- Reportar la información de accounting y facturación.
- Gestionar asociaciones entre los MG y los MGC, permitiendo que un Media Gateway pase a ser controlado por un MGC diferente en caso de que el controlador primario falle.
- Proporcionar una arquitectura flexible y escalable, donde diferentes MGs con diferentes capacidades e interfaces, sean controlados por un único Media Gateway Controller.
- Permitir la actualización, de forma independiente, de los MGs y del MGC. Para ello, se incluye el número de versión a utilizar durante el intercambio de mensajes.

#### 3.1 FUNCIONAMIENTO DEL PROTOCOLO

MGCP se basa en un modelo que contiene endpoints y conexiones.

Los endpoints son fuentes o sumideros de datos multimedia y se encuentran en los media gateways. Dependiendo del tipo de endpoint puede poseer interfaces externas (un endpoint de tipo línea analógica se conecta a una línea analógica).

Una conexión se refiere a la asignación de recursos IP a un endpoint, permitiéndole el acceso a la red. Un endpoint puede tener varias conexiones, por ejemplo, con el fin de realizar conferencias.

Estos recursos IP se describen utilizando el protocolo SDP (Session Description Protocol). Este protocolo se diseñó para describir los parámetros de inicialización de los flujos multimedia, permitiendo la negociación entre los endpoints del formato, tipo de datos y codecs que se van a utilizar en la comunicación.

### 3.1.1 ENDPOINTS MGCP

A continuación se enmueran los tipos de endpoints más importantes:

1. Canal DS0: un canal digital operando a 64Kbps. Habitualmente van multiplexados en DS1 o E1.
2. Línea analógica: es el endpoint necesario para proveer un servicio de telefonía tradicional.
3. Interactive Voice Response (IVR): Sirve para conectar sistemas IVR. En ellos, se reproduce una grabación y, en función de las teclas que pulse el usuario, se realizan unas acciones determinadas.
4. Conference Bridge: Encargado de establecer multiconferencias.
5. Packet relay: Un tipo de Conference Bridge, cuya particularidad es que sólo soporta dos conexiones. Un ejemplo sería un *bouncer* del audio de un endpoint a otro. A modo de analogía con el sistema telefónico antiguo, cuando un usuario llama al operador pidiendo conexión con otro usuario, el operador une los cables para que se realice la llamada. En este modelo, el operador sería el *packet-relay server*.
6. Wiretap: Se hace una conexión desde un punto hasta un endpoint con el fin de escuchar el audio recibido o enviado en ese endpoint. La conexión a un wiretap es unidireccional.
7. ATM trunk-side interface: Correspondiente a una terminación de un enlace ATM. Se puede utilizar en un gateway para proveer una interfaz entre una red VoIP y otra Voice over ATM.

Los endpoints deben estar correctamente identificados para poder acceder a ellos. La identificación se compone del nombre de dominio al que pertenece y un nombre local que depende del gateway en cuestión. Generalmente siguen una estructura W/X/Y/Z, por ejemplo, aaln/S0/SU2/1@Router

### 3.1.2 CONEXIONES MGCP

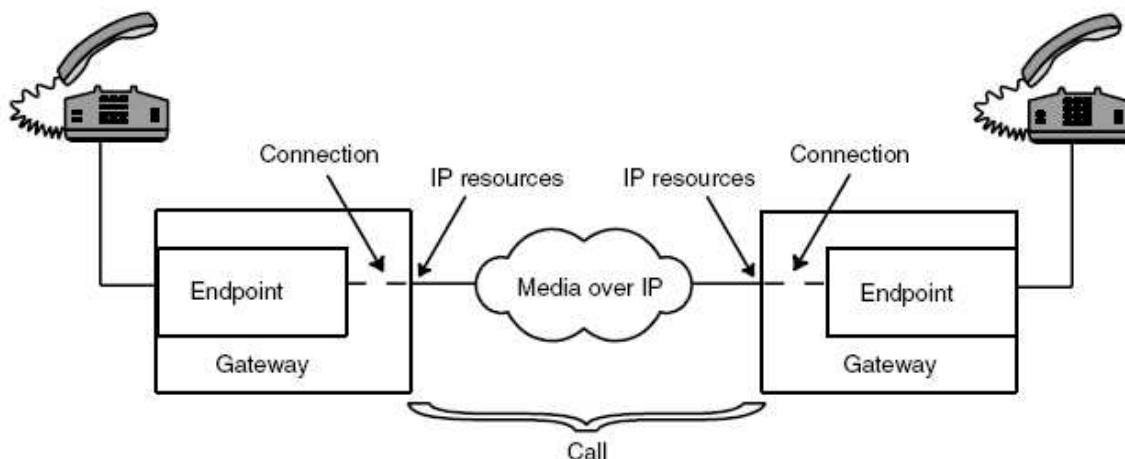
Una conexión es la relación establecida entre un endpoint y una sesión RTP/IP.

Con el fin de estandarizar la realización de las comunicaciones entre MGC y MGs, se han establecido los siguientes puertos predeterminados:

- mgcp-gateway 2427/tcp # Media Gateway Control Protocol Gateway
- mgcp-gateway 2427/udp # Media Gateway Control Protocol Gateway
- mgcp-callagent 2727/tcp # Media Gateway Control Protocol Call Agent
- mgcp-callagent 2727/udp # Media Gateway Control Protocol Call Agent

Aunque se reservan tanto el protocolo de transporte UDP como TCP, el único utilizado para realizar las comunicaciones es el UDP, tal y como se establece en el RFC 3435 relativo a MGCP.

Para efectuar una llamada, se crea un grupo de conexiones de forma que los endpoints asociados con dichas conexiones puedan enviar y recibir datos entre sí. En la siguiente figura se muestra el establecimiento de una llamada entre dos teléfonos analógicos que se comunican mediante la red IP.



**Ilustración 3.1-1. Establecimiento de conexiones para realización de una llamada**

Como se puede observar, en cada gateway se crea una conexión entre el endpoint y los recursos IP. Una vez que ambas conexiones se han creado en los dos Media Gateways, el audio y/o video puede transmitirse produciendo la llamada.

La principal función de MGCP es permitir que se creen dichas conexiones y que se produzca un intercambio de los descriptores de sesiones (SDP) de cada endpoint. Para realizarlo, MGCP utiliza distintos comandos.

### 3.1.3 COMANDOS MGCP

En MGCP se definen nueve comandos. Algunos de ellos se envían del call agent al gateway, y otros en sentido contrario.

Las instrucciones MGCP se componen de una línea de comandos, líneas de parámetros y, opcionalmente, descriptores de sesión.

La línea de comando está formada por 4 elementos separados por un carácter espacio (SP) siguiendo el esquema:

CommandVerb SP TransactionID SP EndpointID SP Version

Siendo:

- CommandVerb: 4 caracteres que sirven para determinar qué comando se envía.
- TransactionID: Identificador de la transacción. Es un identificador único que se define para el seguimiento de las peticiones y respuestas.
- EndpointID: Identificador del endpoint al que se aplica el comando. Es del tipo *endpoint@hostname*, en los routers usados como *Media Gateways*, suelen ser *aaln/S0/SUX/Y@hostname*, con X e Y la tarjeta y el puerto utilizados respectivamente.
- Version: versión del protocolo utilizada (generalmente MGCP 1.0)

Los 9 comandos disponibles son:

- EndpointConfiguration (EPCF): Ejecutado por un call agent para informar al gateway sobre la codificación de uno o más endpoints.
- CreateConnection (CRCX): Ejecutado por un call agent para crear una conexión en el endpoint de un gateway.
- ModifyConnection (MDCX): Ejecutado por un call agent para cambiar algunas de las características de una conexión existente en un gateway. Una de sus funciones principales es informar a un endpoint de los datos de conexión del otro extremo cuando se establece una llamada.
- DeleteConnection (DLCX): Ejecutado por un call agent para terminar una o varias conexiones existentes en un endpoint. También puede ser ejecutado por el gateway para informar al call agent de que el endpoint unido a esa conexión ha perdido la conectividad.
- NotificationRequest (RQNT): Ejecutado por un call agent para solicitar al gateway que le informe de la ocurrencia de ciertos eventos. Le puede indicar, por ejemplo, que detecte cuándo se descuelga un teléfono analógico.
- Notify (NTFY): Ejecutado por un gateway para informar al call agent de la ocurrencia de ciertos eventos.
- AuditEndpoint (AUEP): Ejecutado por un call agent para solicitar información a un gateway sobre un endpoint determinado.
- AuditConnection (AUCX): Ejecutado por un call agent para solicitar información a un gateway sobre una conexión determinada.
- RestartInProgress (RSIP): Ejecutado por un gateway para informar al call agent cuando uno o más endpoints se activan/desactivan.

### 3.1.3.1 Encapsulación de comandos

MGCP soporta el concepto de encapsulación, permitiendo que un comando pueda ser incluido dentro de otro. Por ejemplo, cuando se ordena a un gateway crear una conexión, el call agent puede simultáneamente requerir que se le informe de distintos eventos. Otra utilidad importante es permitir la ejecución de ciertas acciones condicionadas. Un caso práctico sería solicitar al Gateway que detecte y reporte tonos DTMF, pero sólo cuando se está realizando una llamada.

### 3.1.3.2 Líneas de Parámetros MGCP y respuestas a comandos

Cada comando MGCP tiene asociados diferentes parámetros. Se identifican por un código que se compone de una o dos letras mayúsculas. Puede consultarse una lista detallada de los parámetros y de las respuestas a comandos en [1] Carrier Grade Voice Over IP (p.258-274).



### 3.1.4 SEGURIDAD EN MGCP

No existen mecanismos de seguridad implementados en el protocolo MGCP. El RFC 2705 hace referencia al uso de IPsec (AH o ESP) para proteger los paquete. Sin esta protección, un atacante podría establecer llamadas no autorizadas o interferir en el desarrollo de las llamadas establecidas. Además del uso de IPsec, MGCP permite al controlador otorgar claves de sesión a los Media Gateways para cifrar los mensajes de audio, evitando posibles escuchas telefónicas. Las claves de sesión se utilizan para el cifrado del protocolo RTP tal y como se describe en el RFC 1889. Para el intercambio de claves se utiliza el protocolo SDP (RFC 2327).

### 3.1.5 TRAMAS DE EJEMPLO

Es interesante comprobar el funcionamiento del protocolo mediante tramas ethernet capturadas en situaciones clave de la comunicación.

MGCP es un protocolo basado en texto con la siguiente estructura:

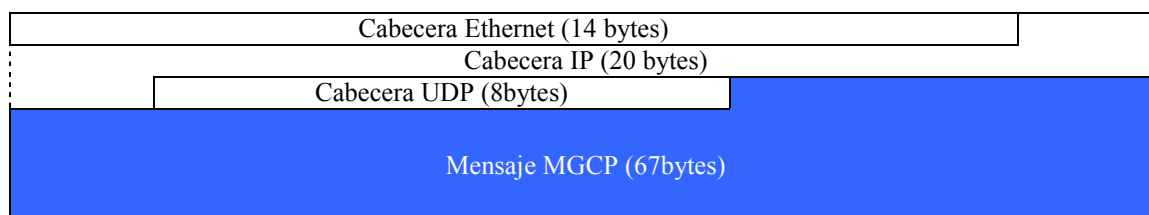
- La primera línea se compone del comando MGCP, el identificador de transacción, el identificador del endpoint y la versión MGCP. Estos campos están separados por espacios y son de tamaño variable, a excepción del comando MGCP que se compone de 4 caracteres.
- Las siguientes líneas son destinadas a parámetros MGCP; Cada línea nueva se corresponde con un parámetro diferente.
- En el caso de existencia de una sesión SDP, se incluiría a continuación de los parámetros MGCP, separada por CRLF-CRLF. Cada campo del protocolo SDP se separa del siguiente mediante un CRLF.

La siguiente figura muestra un paquete capturado mediante wireshark, con los datos relativos a MGCP resaltados:

0000	00 1e 37 2c d9 6a 00 14	22 b0 17 c8 08 00 45 68	..7..j.. ".....Eh
0010	00 5f 00 86 00 00 fa 11	96 8d 0a 03 11 04 0a 06	.....
0020	04 06 09 7b 09 7b 00 4b	56 a3 4e 54 46 59 20 32	...{.k V.NTFY 2
0030	39 20 61 61 6c 6e 2f 53	30 2f 53 55 32 2f 30 40	9 aaln/S 0/SU2/0@
0040	41 52 4d 41 52 49 4f 33	5f 4c 4f 4e 44 4f 4e 20	ARMARIO3 _LONDON
0050	4d 47 43 50 20 31 2e 30	0a 58 3a 20 36 36 61 62	MGCP 1.0 .X: 66ab
0060	35 38 37 36 0a 4f 3a 20	4c 2f 68 64 0a	5876.O: L/hd.

Ilustración 3.1-2. Paquete capturado mediante Wireshark

El paquete capturado muestra la siguiente estructura:





Centrándonos en el Mensaje MGCP de la captura:

		Comando MGCP (4 bytes)	Space	Transaction
ID (2 bytes)	Space	Identificador del Endpoint (29 bytes)		
				Space
Versión MGCP (8bytes)		LF	Parámetro "X" de MGCP (11 bytes)	
	LF	Parámetro "O" de MGCP (7 bytes)		LF

Dado que se trata de un protocolo basado en texto una representación más adecuada sería la siguiente:

```
NTFY 29 aaln/S0/SU2/0@ARMARIO3_LONDON MGCP 1.0
X: 66ab5876
O:L/hd
```

Donde la primera línea se corresponde con el comando MGCP (NTFY), el identificador de la transacción (29), el endpoint al que se hace referencia (aaln/S0/SU2/0@ARMARIO3\_LONDON), y la versión del protocolo (MGCP 1.0). El resto de líneas son parámetros asociados al comando MGCP. Cada parámetro comienza en una línea diferente.

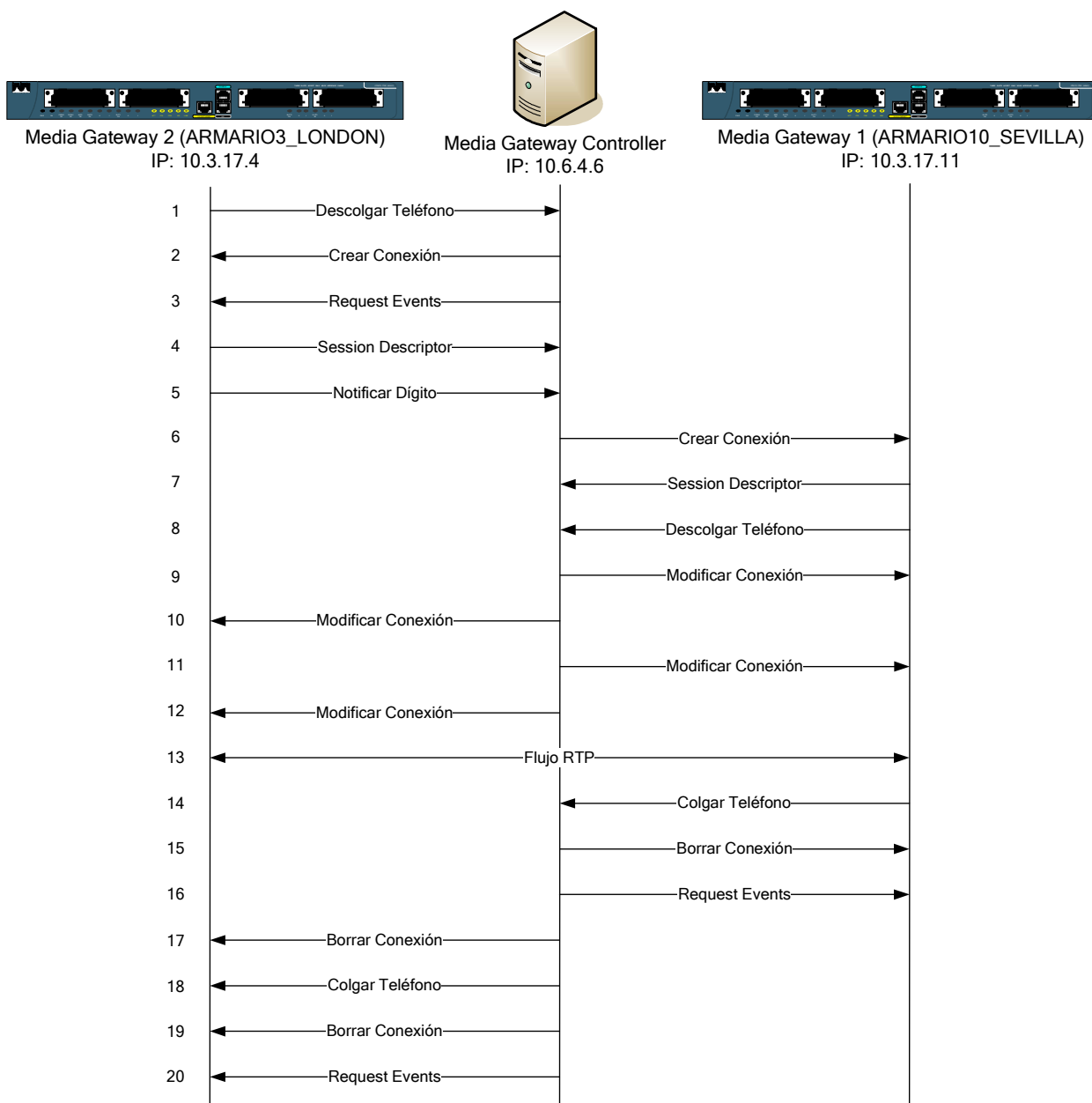
En el RFC 3435 se pueden consultar varios ejemplos de momentos claves en una comunicación MGCP. Los ejemplos expuestos en el presente documento corresponden a pruebas propias realizadas en el laboratorio.

### 3.1.5.1 Establecimiento de llamadas usando MGCP

A continuación se describe el proceso de establecimiento de una llamada entre dos endpoints de dos gateways diferentes, conectados al mismo call agent.

MGCP exige la creación de, al menos, dos conexiones para iniciar una llamada. Descrito en términos generales:

1. El Media Gateway notifica al controlador de que un extremo ha descolgado el teléfono.
2. El controlador ordena al gateway crear una conexión.
3. El controlador solicita el informe de la ocurrencia de ciertos eventos y le ordena que dé tono.
4. El media Gateway responde con los datos de los recursos IP asignados a dicha conexión.
5. El usuario de ese extremo marca el número de teléfono al que desea llamar
6. El call agent requiere al otro gateway la creación de una nueva conexión con los datos IP obtenidos en el paso anterior. Esta conexión se abre en modo sólo lectura.
7. El segundo gateway responde con la información de sus propios recursos IP.
8. Se descuelga el teléfono correspondiente al segundo gateway
9. El controlador ordena al segundo gateway modificar la conexión para poder enviar datos.
10. Se modifica la conexión del primer GW para asignar un identificador de conexión.
11. Se modifica la conexión del segundo GW para reflejar los recursos IP del primer GW
12. Se modifica la conexión del primer GW para reflejar los recursos IP del segundo GW
13. Se realiza el transporte de flujo multimedia entre los dos gateways directamente, sin intervención del call agent.
14. Uno de los dos extremos informa al controlador de que se ha colgado el teléfono.
15. Ante esta situación le ordena que borre la conexión.
16. También le pide que vuelva a notificarle en caso de que el teléfono se descuelgue.
17. Dado que un extremo ha finalizado y ya no queda más gente en esa llamada, el controlador le indica al otro Media Gateway que borre la conexión.
18. El usuario 1, cuelga también el teléfono.
19. El controlador vuelve a solicitar el borrado de la conexión, aunque ya lo haya ordenado anteriormente.
20. Además, le ordena que vuelva a notificarle en caso de que se descuelgue el teléfono.



**Ilustración 3.1-3. Establecimiento de una llamada utilizando el protocolo MGCP**

A continuación se detallan los paquetes involucrados en la llamada:

No.	Time	Source	Destination	Protocol
1	36.446845	10.3.17.4	10.6.4.6	MGCP

**Comando:** NTFY 27 aaln/S0/SU2/0@ARMARIO3LONDON MGCP 1.0

NTFY → Notificar un evento observado

27 → Identificador de la transacción

aaIn/S0/SU2/0@ARMARIO3\_LONDON → Endpoint que notifica la acción

MGCP 1.0 → Versión MGCP

**Parámetros:** O: L/hd

N: ca@10.6.4.6:2427

X: 41b1e77b

O: L/hd → Evento observado: Hook detect (se ha descolgado el teléfono)

N: ca@10.6.4.6:2427 → Entidad a la que se notifica

X: 41b1e77b → Identificador de la petición

No.	Time	Source	Destination	Protocol
2	36.447022	10.6.4.6	10.3.17.4	MGCP/ SDP

**Comando:** CRCX 3326 aaln/S0/SU2/0@ARMARIO3\_LONDON MGCP 1.0

CRCX → Crear conexión

3326

aaIn/S0/SU2/0@ARMARIO3\_LONDON

MGCP 1.0

**Parámetros:** C: 733d34de3ddb2496

L: p:20, a:PCMU

M: sendrecv

X: 3ddb2496

C: 733d34de3ddb2496 → Se establece un identificador de llamada.

L: p:20, a:PCMU → Códec PCMU, paquetes de voz cada 20 ms.

M: sendrecv → Modo de conexión: Enviar y recibir.

X: 3ddb2496 → Identificador de la petición

SDP: v: 0  
o: root 19129 19129 IN IP4 10.6.4.6  
s: session  
c: IN IP4 10.6.4.6  
m: audio 13064 RTP 0  
a: rtpmap:0 PCMU/8000  
X: 41b1e77b

V: 0 → Versión del protocolo SDP

O: root 19129 19129 IN IP4 10.6.4.6

Usuario propietario: root  
Identificador de sesión: 19129  
Versión de la sesión: 19129  
Tipo de red: IN (Internet)  
Tipo de dirección: IP4 (IPV4)  
IP: 10.6.4.6

S: session → Nombre de la sesión: *session*

C: IN IP4 10.6.4.6 → Información de la conexión

M: audio 13064 RTP 0 { Tipo de datos multimedia: *audio*  
Puerto por el que se realiza la comunicación: *13064*  
Protocolo: RTP  
Código: ITU-T G.711 PCMU

A: rtpmap:0 PCMU/8000 { Atributos de la sesión. Tasa de muestreo 8000 para el  
código G.711

X: 41b1e77b → Identificador de la petición

No.	Time	Source	Destination	Protocol
3	36.447041	10.6.4.6	10.3.17.4	MGCP

Comando: RQNT 3327 aaln/S0/SU2/0@ARMARIO3\_LONDON MGCP 1.0

RQNT → Solicitar la notificación de la ocurrencia de ciertos eventos

Parámetros: R: L/lu(N),L/hf(N),D/[0-9#\*](N)

S: L/dl

X: 41b1e77b

R: L/lu(N),L/hf(N),D/[0-9#\*](N)

El MGC solicita al MG que le informe si se cuelga el teléfono, si se hace un *hook-flash* o si se marca cualquier dígito.

S: L/dl → Tono de llamada en el teléfono

No.	Time	Source	Destination	Protocol
4	36.512075	10.3.17.4	10.6.4.6	MGCP

Comando: 200 3326 OK

200 → Confirmación de que se ha realizado correctamente una transacción.

3326 → Transacción que se ha realizado

OK → Resultado: OK

SDP: v: 0

o: - 9 0 IN IP4 10.3.17.4

s: Cisco SDP 0

c: IN IP4 10.3.17.4

m: audio 16760 RTP 0 100

a: rtpmap:100 X-NSE/8000

No.	Time	Source	Destination	Protocol
5	37.885124	10.3.17.4	10.6.4.6	MGCP

Comando: NTFY 28 aaln/S0/SU2/0@ARMARIO3\_LONDON MGCP 1.0

Parámetros

N: ca@10.6.4.6:2427

O: D/2 → Se ha pulsado el dígito 2

X: 41b1e77b

Existen otros paquetes similares indicando la pulsación del resto de dígitos.

No.	Time	Source	Destination	Protocol
6	42.940434	10.6.4.6	10.3.17.11	MGCP

Comando: CRCX 3341 aaln/S0/SU2/0@ARMARIO10\_SEVILLA MGCP 1.0

Parámetros: C: 081e2f7e289b8af7

L: p:20, a:PCMU

M: recvonly

X: 289b8af7

M: recvonly → Modo de conexión: Únicamente recibir.

SDP: v: 0

o: root 19129 19129 IN IP4 10.6.4.6

s: session

c: IN IP4 10.6.4.6

m: audio 18920 RTP 0

a: rtpmap:0 PCMU/8000

No.	Time	Source	Destination	Protocol
7	43.011498	10.3.17.11	10.6.4.6	MGCP

Comando: 200 3341 OK

SDP: v: 0  
o: - 16 0 IN IP4 10.3.17.11  
s: Cisco SDP 0  
c: IN IP4 10.3.17.11  
m: audio 18604 RTP 0 100  
a: rtpmap:100 X-NSE/8000  
t: 0 0

T: 0 0 → Tiempo de sesión

No.	Time	Source	Destination	Protocol
8	46.808231	10.3.17.11	10.6.4.6	MGCP

Comando: NTFY 65 aaln/S0/SU2/0@ARMARIO10\_SEVILLA MGCP 1.0

Parámetros: C: 081e2f7e289b8af7  
O: L/hd  
X: 1a2fa5a3

Se notifica que ha sido descolgado el teléfono de destino.

No.	Time	Source	Destination	Protocol
9	46.808312	10.6.4.6	10.3.17.11	MGCP

Comando: MDCX 3344 aaln/S0/SU2/0@ARMARIO10\_SEVILLA MGCP 1.0

Parámetros: C: 081e2f7e289b8af7  
M: sendrecv  
R: L/hu(N),L/hf(N),D/[0-9#\*](N)

Se modifica la conexión del teléfono que ha sido descolgado para que también pueda enviar datos.

No.	Time	Source	Destination	Protocol
10	46.809414	10.6.4.6	10.3.17.4	MGCP

**Comando:** MDCX 3346 aaln/S0/SU2/0@ARMARIO3\_LONDON MGCP 1.0

**Parámetros:** C: 081e2f7e289b8af7  
M: sendrecv  
R: L/hu(N),L/hf(N),D/[0-9#\*](N)  
I: 9

I: 9 → Identificador de la conexión

Se actualiza la conexión original para solicitarle información sobre la ocurrencia de ciertos eventos y establecer un identificador de conexión.

No.	Time	Source	Destination	Protocol	Info
11	46.847247	10.6.4.6	10.3.17.11	MGCP	MGCP/SDP MDCX 3350 aaln/S0/SU2/0@ARMARIO10_SEVILLA A MGCP 1.0, with session description

**Comando:** MDCX 3350 aaln/S0/SU2/0@ARMARIO10\_SEVILLA MGCP 1.0

**Parámetros:** C: 081e2f7e289b8af7  
L: p:20, a:PCMU  
M: sendrecv  
R: L/hu(N),L/hf(N),D/[0-9#\*](N)  
X: 289b8af7

**SDP:** v: 0  
o: root 19129 19129 IN IP4 10.3.17.4  
s: session  
c: IN IP4 10.3.17.4  
m: audio 16760 RTP 0  
a: rtpmap:0 PCMU/8000

Se informa al Gateway ARMARIO10\_SEVILLA, que el otro extremo se encuentra en la IP 10.3.17.4 correspondiente al ARMARIO3\_LONDON



No.	Time	Source	Destination	Protocol	Info
12	46.950451	10.6.4.6	10.3.17.4	MGCP	MGCP/SDP MDCX 3349 aaln/S0/SU2/0@ARMARIO3_LONDON MGCP 1.0, with session description

Comando: MDCX 3349 aaln/S0/SU2/0@ARMARIO3\_LONDON MGCP 1.0

Parámetros: C: 733d34de3ddb2496  
L: p:20, a:PCMU  
M: sendrecv  
R: L/hu(N),L/hf(N),D/[0-9#\*](N)  
X: 3ddb2496  
I: 9

SDP: v: 0  
o: root 19129 19129 IN IP4 10.3.17.11  
s: session  
c: IN IP4 10.3.17.11  
m: audio 16502 RTP 0  
a: rtpmap:0 PCMU/8000

En este caso es al gateway ARMARIO3\_LONDON a quien se le dice quién es el extremo de la conexión, quedando ya la conexión punto a punto establecida.

No.	Time	Source	Destination	Protocol
13	NA (otra captura)	10.3.17.4	10.3.17.11	RTP

La comunicación RTP se ha establecido entre ambos nodos. Los datos de voz van directamente de un extremo a otro sin pasar por el MGC.

No.	Time	Source	Destination	Protocol
14	120.07739	10.3.17.11	10.6.4.6	MGCP

Comando: NTFY 33 aaln/S0/SU2/0@ ARMARIO10\_SEVILLA MGCP 1.0

Parámetros: N: ca@10.6.4.6:2427  
X: 543c6cca  
O: L/hu  
O: L/hu → Se ha colgado un extremo

El teléfono correspondiente al gateway ARMARIO10\_SEVILLA, se ha colgado. Este hecho se le notifica al controlador para que realice las acciones pertinentes.

No.	Time	Source	Destination	Protocol
15	120.08547	10.6.4.6	10.3.17.11	MGCP

Comando: DLCX 3359 aaln/S0/SU2/0@ ARMARIO10\_SEVILLA MGCP 1.0  
DLCX → Borrar conexión

Parámetros: C: 081e2f7e289b8af7  
X: 289b8af7  
I: A

El controlador, tras recibir el paquete que le informaba del colgado de un extremo, procede a ordenar la eliminación de la conexión correspondiente a dicho extremo.

No.	Time	Source	Destination	Protocol
16	120.08549	10.6.4.6	10.3.17.11	MGCP

Comando: RQNT 3360 aaln/S0/SU2/0@ ARMARIO10\_SEVILLA MGCP 1.0

Parámetros: X: 543c6cca  
R: L/hd(N)

El Gateway vuelve a la situación inicial de informar en caso de que se descuelgue el teléfono.

No.	Time	Source	Destination	Protocol
17	120.08561	10.6.4.6	10.3.17.4	MGCP

Comando: DLCX 3362 aaln/S0/SU2/0@ ARMARIO3\_LONDON MGCP 1.0

Parámetros: C: 733d34de3ddb2496  
X: 3ddb2496  
I: 9

También es necesario borrar la conexión del otro punto, ya que la llamada se ha dado por finalizada.

No.	Time	Source	Destination	Protocol
18	131.85032	10.3.17.4	10.6.4.6	MGCP

Comando: NTFY 32 aaln/S0/SU2/0@ARMARIO3\_LONDON MGCP 1.0

Parámetros: N: ca@10.6.4.6:2427  
X: 41b1e77b  
O: L/hu

También se ha colgado este teléfono.

No.	Time	Source	Destination	Protocol	Info
19	131.85061	10.6.4.6	10.3.17.4	MGCP	DLCX 3367 aaln/S0/SU2/0@ARMARIO3_LONDON MGCP 1.0

Comando: DLCX 3367 aaln/S0/SU2/0@ ARMARIO3\_LONDON MGCP 1.0

Parámetros: X: 3ddb2496

Vuelve a solicitar el borrado de la conexión una vez colgado el teléfono. A lo que el gateway le responde con un *250 already deleted connection*, ya que había sido borrada anteriormente.

No.	Time	Source	Destination	Protocol
20	131.85063	10.6.4.6	10.3.17.4	MGCP

Comando: RQNT 3368 aaln/S0/SU2/0@ARMARIO3\_LONDON MGCP 1.0

Parámetros: X: 41b1e77b  
R: L/hd(N)

Por último, este endpoint vuelve a su estado inicial.

### 3.1.5.2 CONFERENCIA

Cuando se realiza una conferencia es el servidor Asterisk el que gestiona todo el tráfico de voz. Cada cliente envía su audio al controlador y éste se encarga de redistribuirlo a los demás nodos. El esquema para la inicialización y finalización de llamada es similar al explicado anteriormente. En formato digital se incluyen las trazas correspondientes a una conferencia (*directorio/capturas/conferencia/*).

### 3.1.5.3 ENCENDIDO DE UN MEDIA GATEWAY

No.	Time	Source	Destination	Protocol
1	0.000000	10.3.17.4	10.6.4.6	MGCP

Comando: RSIP 1 \*@ARMARIO3\_LONDON MGCP 1.0

RSIP → RestartInProgress (Se está produciendo un restart del gateway/endpoint)

\*@ARMARIO3\_LONDON → Todos endpoints del Gateway ARMARIO3\_LONDON

Parámetros: RM:restart

RM:restart → Evento observado: Hook detect (se ha descolgado el teléfono)

El Media Gateway informa al controlador de que se está reiniciando.

No.	Time	Source	Destination	Protocol
2	0.000054	10.6.4.6	10.3.17.4	MGCP

Comando: 200 1 OK

200 → Confirmación de que se ha realizado correctamente una transacción.

1 → Transacción que se ha realizado

OK → Resultado: OK

No.	Time	Source	Destination	Protocol
3	0.000067	10.6.4.6	10.3.17.4	MGCP

Comando: RQNT 189 \*@ARMARIO3\_LONDON MGCP 1.0

Parámetros: X: 0c80025f

R: L/hd(N)

El controlador pide que se le notifique en caso de que se descuelgue el teléfono.

No.	Time	Source	Destination	Protocol
4	0.000074	10.6.4.6	10.3.17.4	MGCP

Comando: AUEP 190 \*@ARMARIO3\_LONDON MGCP 1.0

AUEP → Auditar un endpoint

Parámetros: F: A

F: A → Solicita los identificadores de todos endpoints

No.	Time	Source	Destination	Protocol
5	0.025621	10.3.17.4	10.6.4.6	MGCP

Comando: 200 190

200 → Transacción correcta. OK.

190 → Identificador de la transacción

Parámetros: Z: aaln/S0/SU2/0@ARMARIO3\_LONDON

Z: aaln/S0/SU2/1@ARMARIO3\_LONDON

Z: aaln/S0/SU2/0@ARMARIO3\_LONDON

Z: aaln/S0/SU2/1@ARMARIO3\_LONDON → Identificadores de los endpoints

### 3.1.5.4 APAGADO CONTROLADO DE UN MEDIA GATEWAY

No.	Time	Source	Destination	Protocol
1	0.000000	10.3.17.4	10.6.4.6	MGCP

Comando

RSIP

40

aaln/S0/SU2/0@ARMARIO3\_LONDON

MGCP 1.0

Parámetros

RM: graceful { Reinicio *graceful* (Las conexiones existentes se siguen manteniendo, pero el controlador no debe iniciar nuevas conexiones que afecten al gateway indicado.

RD: 0 → Tiempo hasta el reinicio

### 3.1.5.5 APAGADO FORTUITO DE UN MEDIA GATEWAY

Cuando un Media Gateway se apaga de forma inesperada, el controlador no es notificado de que dicho Media Gateway está inoperativo. El Controlador no tendrá constancia de dicho problema hasta que intente realizar alguna conexión con un endpoint correspondiente al Media Gateway.

## 4. ASTERISK

### 4.1 INTRODUCCIÓN AL SOFTWARE Y A SU FUNCIONALIDAD

Asterisk es una aplicación de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX). Como en cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o a una línea analógica tradicional o RDSI.

Asterisk incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática de llamadas, etc. Los usuarios pueden crear nuevas funcionalidades escribiendo un dialplan en el lenguaje de script de Asterisk o añadiendo módulos escritos en cualquier lenguaje de programación soportado por Linux.

Una de las características más interesantes de Asterisk es la gran cantidad de protocolos VoIP que soporta, como pueden ser SIP, H.323, IAX y MGCP.

### 4.2 CONFIGURACIÓN

En esta sección se van a describir los pasos necesarios para configurar el software de Asterisk con todas las funcionalidades necesarias. Los archivos de configuración completos se pueden encontrar íntegramente en el apéndice correspondiente.

#### 4.2.1 SERVIDOR ASTERISK

El fichero que define los parámetros a nivel de ejecución usados por el demonio es: */etc/asterisk/asterisk.conf*. Es importante especificar los directorios donde se van a guardar los archivos utilizados por Asterisk, así como el usuario y grupo que lo va a ejecutar (hacerlo como root es peligroso y altamente desaconsejado).

*/etc/asterisk/asterisk.conf*

[directories]

astetcdir => /etc/asterisk ; Directorio donde se encuentran los archivos de configuración.

astmoddir => /usr/lib/asterisk/modules ; Directorio para los módulos de Asterisk.

astvarlibdir => /var/lib/asterisk

astdbdir => /var/lib/asterisk

astkeydir => /var/lib/asterisk

astdatadir => /var/lib/asterisk

astagidir => /var/lib/asterisk/agi-bin ; Directorio donde se guardan los scripts a ejecutar ; por AGI.

astspooldir => /var/spool/asterisk

astrundir => /var/run/asterisk ; Directorio para guardar el archivo *pid* de Asterisk.

astlogdir => /var/log/asterisk ; Directorio para guardar los logs de Asterisk.

**[options]**

runuser = asterisk ; Usuario que va a ejecutar Asterisk.

rungroup = asterisk ; Grupo que va a ejecutar Asterisk.

**[compat]**

pbx\_realtime=1.6

res\_agi=1.6

app\_set=1.6

NOTA: El resto de opciones se pueden mantener comentadas o borrarlas.

**4.2.2 MGCP**

Asterisk dispone de un soporte muy básico para el protocolo MGCP. Esta limitación ha ocasionado diversos problemas que se solventaron posteriormente. La configuración relativa al protocolo MGCP se encuentra en el archivo */etc/asterisk/mgcp.conf*

Este fichero se compone de una sección general compartida por todos los Media Gateways y de secciones específicas para cada host. En la sección general se indican parámetros globales como el puerto y dirección IP en el que escucha el Media Gateway Controller.:

**[general]**

*port = 2427 ; Puerto en el que escucha el MGC*

*bindaddr = 10.6.4.6 ; IP en la que escucha*

*disallow = all ; Desactivamos todos los codecs*

*allow = g726 ; Permitimos el g726*

*allow = alaw ; Permitimos G711.a*

*allow = ulaw ; Permitimos G711.u*

*;allow = gsm (NO SE PUEDE PERMITIR PORQUE EL MEDIA GATEWAY DA COMUNICANDO)*

*allow = speex*

*tos = af31 ; Asignamos la prioridad DSCP para los paquetes de señalización*

*tos\_audio = ef ; Asignamos la prioridad DSCP para los paquetes de voz.*

Además, cada host al que queramos otorgar acceso debe tener una sección similar a esta:

*[ARMARIO1] ; Nombre del host.*

*callwaiting=no ; Deshabilitamos la llamada en espera.*

*threewaycalling=yes ; Habilitamos la llamada a 3.*

*transfer=yes ; Habilitamos la transferencia de llamadas.*

*host = 10.4.88.2 ; IP del host a la que se va a dar acceso.*

*context = internal ; contexto dentro del plan de llamadas al que se le asigna.*

*callerid= " ARM1\_1 " ; Identificador de llamada.*

*line => aaln/S0/SU2/0 ; Endpoint 1 asignado en el Media Gateway.*

*callerid= " ARM1\_2 "*

*line => aaln/S0/SU2/1 ; Endpoint 2 asignado en el Media Gateway.*

*line => \* ; Parámetro necesario para que asterisk pueda acceder a todos los endpoints.*

### 4.2.3 SIP

En la actualidad, existen dos servidores SIP en el Laboratorio de Telemática ofreciendo servicio para ese protocolo, por ello, en el presente proyecto no se preveía su implementación. No obstante, diversos problemas con los teléfonos IP-300 causaron que, finalmente se optase por su soporte.

Para su utilización en Asterisk hay que configurar el fichero */etc/asterisk/sip.conf*. Su estructura es similar a la del fichero correspondiente a MGCP, con un apartado general y uno específico para cada cliente:

```
[general]
context=internal
srvlookup=yes
canreinvite=yes
disallow=all
allow=ulaw
allow=g726
autocreatepeer=yes
```

```
[SIP1]
type=friend
secret=Sip1
qualify=yes
nat=no
host=dynamic
canreinvite=yes
context=internal
callwaiting=yes
dtmf=rfc2833
```

### 4.2.4 DAHDI

El uso de las tarjetas pci de telefonía se regula mediante el fichero *chan\_dahdi.conf*, configurando, principalmente, la señalización, el contexto y los codecs a utilizar .

```
[trunkgroups]
[channels]
disallow = all
allow = ulaw
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
echocancel=yes
echotraining=yes
immediate=no
```



;A continuación se definen los canales y contextos

*context=phones* ; Las llamadas a realizar desde los teléfonos analógicos los introducimos en el contexto internal

*signalling=fxo\_ks* ; Señalización FXO kewlstart para los puertos FXS

*channel => 3* ; Esta configuración se le aplica a los canales 3 y 4.

*channel => 4*

*context=incoming\_calls*; Aquí configuramos las llamadas entrantes para que vayan a un contexto diferente

*signalling=fxs\_ks*

*channel => 2*

## 4.2.5 PLAN DE LLAMADAS

El plan de llamadas (Dial plan) es el núcleo de Asterisk. Define cómo va a manejar Asterisk cada llamada entrante o saliente. El fichero de configuración contiene extensiones que forman una lista de instrucciones que Asterisk debe seguir. La mayor parte del *dial plan* se encuentra en el archivo */etc/asterisk/extensions.conf*.

Este fichero se compone, básicamente, de cuatro conceptos:

- Extensiones
- Prioridades
- Aplicaciones
- Contextos

### 4.2.5.1 Estructura del fichero *extensions.conf*

El fichero *extensions.conf* se divide en secciones. La primera es la *[general]* seguida por la sección *[global]*. El inicio de cada sección comienza con la definición de su nombre *[nombre\_de\_la\_sección]* y finaliza cuando otra sección es creada.

- La sección *[general]*  
Se encuentra situada al inicio del fichero. Antes de configurar el plan de llamadas, es recomendable conocer las opciones generales que controlan algunos comportamientos del dial plan:
  1. *Static y write protect*: Controlan que se permita usar el CLI de Asterisk para sobrescribir el plan de llamadas.
  2. *Autofallthrough*: Si está activado, cuando una extensión finaliza las instrucciones a realizar, Asterisk terminará la llamada con un BUSY, CONGESTION o HANGUP según crea conveniente.
  3. *Clearglobalvars*: Si está activado, las variables globales se borrarán cuando se recargue el plan de llamadas o se reinicie Asterisk.
  4. *Userscontext*: Es el contexto por defecto al que se van a unir los usuarios correspondientes al *users.conf*
- La sección *[globals]*  
Define las variables globales y sus valores iniciales. Se puede acceder a la variable desde cualquier contexto del plan de llamadas mediante *\$\_GLOBAL(variable)*. Incluso se puede acceder a variables definidas en el entorno de linux usando *\$\_ENV(variable)*.

- Contextos

Después de las secciones *[general]* y *[globals]*, el plan de llamadas se compone de contextos. Cada contexto posee a su vez diversas extensiones, permitiendo agrupar los usuarios por contextos y definir los accesos a distintos tipos de llamadas.

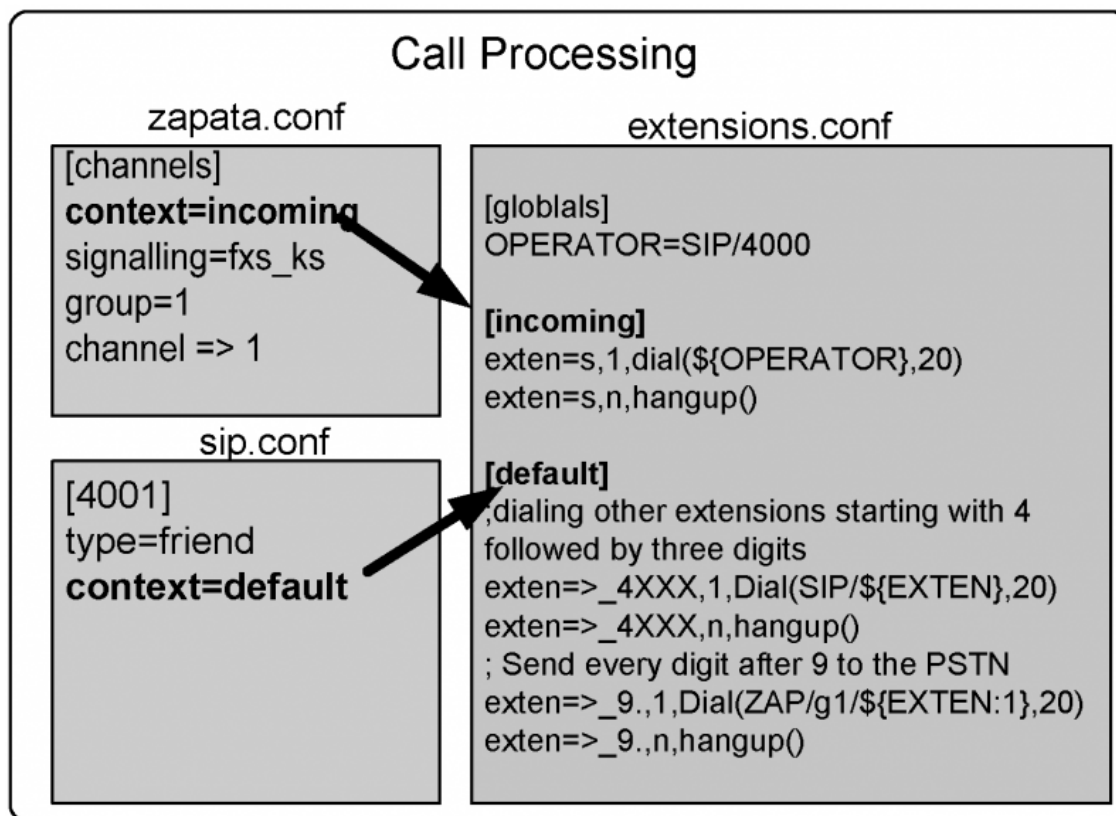


Ilustración 4.2-1. Influencia de los contextos en el procesamiento de las llamadas

- EXTENSIONES

A diferencia de los sistemas tradicionales PBX donde las extensiones están asociadas con teléfonos, menús, etc. En Asterisk una extensión es una lista de comandos a ser ejecutados. Estas instrucciones son procesadas según su orden de prioridad.

La sintaxis de una extensión es:

`exten=> número_de_la_extensión, prioridad, aplicación a ejecutar.`

Número\_de\_la\_extensión se corresponde al número que marca un cliente.

Prioridad: Empieza en 1 y se requiere números sucesivos. Se puede usar prioridad “n” para indicar el número siguiente.

Aplicación: La aplicación a ejecutar. Las aplicaciones disponibles se comentan posteriormente.

Existen extensiones especiales:

- s: Start. Es la extensión por defecto cuando no hay un número que llama. Es especialmente útil en los enlaces FXO.
- t: Timeout. Se usa cuando una llamada ha estado inactiva en espera de una pulsación de tecla, por ejemplo, en un sistema de respuesta de voz interactivo.

- T: Absolute Timeout. Se puede establecer un límite máximo de tiempo por llamada usando la función `absolutetimeout()`. Cuando la llamada excede dicho límite, se envía a la extensión T.
- h: Hangup. Se le llama una vez que un usuario ha desconectado la llamada.
- i: Invalid. Se llega a ella si se marca una extensión inexistente en el contexto.
- APLICACIONES  
Las aplicaciones son los comandos que se ejecutan en las extensiones. Las aplicaciones disponibles se encuentran implementadas en distintos módulos y se pueden consultar mediante el *Command Line Interface*(CLI) de Asterisk. Una vez situado en el CLI (*asterisk -rvvv*), se debe ejecutar:  
*CLI>core show applications*

A continuación se destacarán las 5 aplicaciones más básicas:

- Answer: Contesta un canal si están llamando.
- Dial: Es el comando utilizado para llamar. Se detallará en el siguiente apartado debido a su amplia extensión e importancia en el plan de llamadas.
- Hangup: Esta aplicación se encarga de colgar un canal.
- Goto([[contexto|]extensión|]prioridad): Esta aplicación hace que la llamada continúe su ejecución en el contexto, extensión y prioridad establecidos. Si el intento de saltar a otra parte del plan de llamadas no es satisfactorio, la llamada proseguirá en la siguiente prioridad de la extensión actual.
- Playback: Reproduce un fichero de audio.

#### 4.2.5.2 La aplicación Dial.

Esta aplicación es una de las más importantes de las que se dispone en el plan de llamadas. Es la encargada de llamar a uno o varios canales.

La sintaxis general es la siguiente:

- Llamar a un único canal  
*Dial(tipo/identificador,timeout,opciones)*
- Llamar a múltiples canales  
*Dial(Tipo1/recurso1[&Tipo2/recurso2...][|timeout][|opciones]):*

A menos que se haya especificado un timeout, la aplicación Dial esperará indefinidamente hasta que uno de los canales conteste, el llamante cuelgue o todos los canales se encuentren ocupados o no disponibles.

La aplicación establece las siguientes variables una vez que se ha completado su ejecución:

- DIALEDTIME: El tiempo que transcurre desde que se llama a un canal hasta que se desconecta
- ANSWEREDTIME: Duración de la llamada actual.
- DIALSTATUS: Informa del estado de la llamada.
  - CHANUNAVAIL: Canal no disponible. Ocurre si el extremo no se ha registrado o no se puede acceder a él.
  - CONGESTION: Congestión. Estado debido, normalmente, a que el número al cual se ha llamado no se reconoce.
  - NOANSWER: No se ha contestado. El teléfono destino sonó varios rings y finalmente la aplicación Dial ha dado un timeout.

- BUSY: Señal de ocupado. El destinatario se encontraba comunicando.
- ANSWER: La llamada se ha contestado. Se ha conseguido conectar al llamante con el destino deseado.
- CANCEL: La llamada ha sido cancelada. Se alcanzó el número de destino, pero el llamante colgó antes de que nadie respondiese.
- DONTCALL: Modo privacidad. El número de destino rechazó la llamada.
- TORTURE: Modo privacidad. El destino ha decidido enviar la llamada al menú “torture”.
- INVALIDARGS: Se ha producido un error al enviar los argumentos de la aplicación Dial. Suele ocurrir por errores de sintaxis en el plan de llamadas.

Sobre las opciones disponibles a la hora de ejecutar el comando Dial, vamos a destacar las siguientes:

- r – Indica tono de llamada a la parte llamante. No transmite el audio hasta que el otro extremo contesta.
- t – Permite a la parte llamada transferir la llamada.
- T – Permite a la parte llamante transferir la llamada.
- W – Permite a la parte llamante activar la grabación de la llamada.
- W – Permite a la parte llamada activar la grabación de la llamada.
- k – Permite a la parte llamada aparcas la llamada actual.
- K – Permite a la parte llamante aparcas la llamada actual.

Se pueden consultar en detalle las opciones del comando Dial en: <http://www.asterisk.org/docs/asterisk/trunk/applications/dial?type=applications&value=Dial>

#### 4.2.5.3 Plan de llamadas configurado.

El plan de llamadas configurado en Asterisk para la realización del presente proyecto es el siguiente:

[globals]

LINEA=DAHDI/2-1 ;Establecemos la variable global LINEA para que apunte a uno de los puertos de la tarjeta de telefonía instalada en el PC.

[general]

[incoming\_calls]; Contexto utilizado para las llamadas entrantes. Contesta el canal y ejecuta la aplicación Echo(repita lo que el usuario dice).

exten => s,1,Answer()

exten => s,2,Echo()

;Los siguientes contextos son para la realización de llamadas externas mediante la línea analógica. Se han separado en tipo de llamadas (provinciales, interprovinciales, a números 90X, números 80X y móviles). Para poder realizar estas llamadas, hay que conectar una línea analógica externa en el puerto 2-1 de la tarjeta pci instalada en el PC que hace de servidor de Asterisk.

## [provinciales\_analogico]

```
exten => _9948XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
```

```
exten => _9948XXXXXXX,2,Congestion()
```

```
exten => _9948XXXXXXX,102,Congestion()
```

## [interprovinciales\_analogico]

```
exten => _99ZXXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
```

```
exten => _99ZXXXXXXX,2,Congestion()
```

```
exten => _99ZXXXXXXX,102,Congestion()
```

## [90X\_analogico]

```
exten => _990XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
```

```
exten => _990XXXXXXX,2,Congestion()
```

```
exten => _990XXXXXXX,102,Congestion()
```

## [80X\_analogico]

```
exten => _980XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
```

```
exten => _980XXXXXXX,2,Congestion()
```

```
exten => _980XXXXXXX,102,Congestion()
```

## [moviles\_analogico]

```
exten => _96XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
```

```
exten => _96XXXXXXX,2,Congestion()
```

```
exten => _96XXXXXXX,102,Congestion()
```

## [todas\_analogico]

```
exten => _9XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
```

```
exten => _9XXXXXXX,2,Congestion()
```

```
exten => _9XXXXXXX,102,Congestion()
```

;Los siguientes contextos son para la realización de llamadas externas mediante la  
;maqueta SIP existente en el Laboratorio de Telemática. Se han organizado de manera  
;similar a los contextos analógicos.

## [provinciales\_sip]

```
exten => _0948XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
```

```
exten => _0948XXXXXXX,2,Hangup()
```

```
exten => _0948XXXXXXX,102,Hangup()
```

## [interprovinciales\_sip]

```
exten => _09ZXXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
```

```
exten => _09ZXXXXXXX,2,Hangup()
```

```
exten => _09ZXXXXXXX,102,Hangup()
```

## [90X\_sip]

```
exten => _090XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
```

```
exten => _090XXXXXXX,2,Hangup()
```

```
exten => _090XXXXXXX,102,Hangup()
```

[80X\_sip]

exten => \_080XXXXXXXX,1,Dial(SIP/10.6.5.250/\${EXTEN})

exten => \_080XXXXXXXX,2,Hangup()

exten => \_080XXXXXXXX,102,Hangup()

[moviles\_sip]

exten => \_06XXXXXXXXXX,1,Dial(SIP/10.6.5.250/\${EXTEN})

exten => \_06XXXXXXXXXX,2,Hangup()

exten => \_06XXXXXXXXXX,102,Hangup()

[todas\_sip]

exten => \_0XXXXXXXXXX,1,Dial(SIP/10.6.5.250/\${EXTEN})

exten => \_0XXXXXXXXXX,2,Hangup()

exten => \_0XXXXXXXXXX,102,Hangup()

;La siguiente sección es la más larga de todas. En ella se definen todos los números, a nivel interno, que se usan en la maqueta.

;En primer lugar, comentar que el rango de numeración establecido para esta maqueta es el 5XXX.

[internal]

; NUMERACION CORRESPONDIENTE A LOS ARMARIOS

; Se ha decidido que la numeración sea 5XXY

; Siendo XX el numero de armario

; Siendo Y 1 o 2, en función del puerto al que queramos llamar.

exten => 5011,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO1,rt)

exten => 5012,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO1,rt)

exten => 5021,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO2,rt)

exten => 5022,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO2,rt)

exten => 5031,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO3,rt)

exten => 5032,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO3,rt)

exten => 5041,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO4,rt)

exten => 5042,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO4,rt)

exten => 5051,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO5,rt)

exten => 5052,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO5,rt)

exten => 5061,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO6,rt)

exten => 5062,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO6,rt)

exten => 5071,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO7,rt)

exten => 5072,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO7,rt)

exten => 5081,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO8,rt)

exten => 5082,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO8,rt)

```
exten => 5091,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO9,rt)
exten => 5092,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO9,rt)
```

```
exten => 5101,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO10,rt)
exten => 5102,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO10,rt)
```

```
exten => 5111,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO11,rt)
exten => 5112,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO11,rt)
```

```
exten => 5121,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO12,rt)
exten => 5122,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO12,rt)
```

```
exten => 5131,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO13,rt)
exten => 5132,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO13,rt)
```

```
exten => 5141,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO14,rt)
exten => 5142,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO14,rt)
```

```
exten => 5151,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO15,rt)
exten => 5152,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO15,rt)
```

;Los siguientes dos contextos definen dos tipos de usuario a la hora de permitir realizar llamadas. Estos contextos se pueden asignar en los ficheros *mgcp.conf* y *sip.conf* para que ciertos usuarios puedan realizar todas las llamadas y permitir, a otro tipo de usuarios, la realización de llamadas internas exclusivamente.

[profesores] ;Pueden realizar llamadas a cualquier número y por ello, se incluyen todos contextos anteriores.

```
include => todas_analogico
```

```
include => todas_sip
```

```
include => internal
```

[alumnos] ;Sólo pueden realizar llamadas internas.

```
include => internal
```

### 4.3 CALL DETAIL RECORD

En un sistema de telefonía es esencial contar con un buen registro de detalle de llamadas. Esta información es muy útil, tanto si se desea facturar las llamadas realizadas como para efectuar un seguimiento del uso de la central telefónica.

Asterisk soporta el uso de CDR, pudiendo almacenar la información en cualquiera de los siguientes formatos:

- Csv – archivos de texto con valores separados por comas.
- Cdr SQLite – logs de CDR en la base de datos de SQLite.
- Pgsq – logs de CDR en las bases de datos de PostgreSQL.
- Odbc – logs de CDR a cualquier base de datos con soporte unixODBC.
- Mysql – logs de CDR en las bases de datos de MySQL.
- Cdr FreeTDS – logs CDR en MS SQL o a la base de datos de Sybase a través de los conductores de FreeTDS
- Yada – logs de CDR registrados en cualquier base de datos con soporte yada.



### 4.3.1 CREACIÓN DE LA BASE DE DATOS

Los registros de detalle de llamadas se van a almacenar en una base de datos Mysql alojada en el servidor Jupiter (10.1.1.230). Los pasos para su creación son los siguientes:

```
CREATE DATABASE CDRsFromMGC;

GRANT INSERT
  ON CDRsFromMGC.*
  TO mgc@10.6.4.6
  IDENTIFIED BY '2202621450';
GRANT INSERT
  ON CDRsFromMGC.*
  TO mgc@ 10.6.4.201
  IDENTIFIED BY '2202621450';
GRANT INSERT
  ON CDRsFromMGC.*
  TO mgc@ 10.6.4.202
  IDENTIFIED BY '2202621450';

USE CDRsFromMGC;
CREATE TABLE cdr (
  calldate datetime NOT NULL default '0000-00-00 00:00:00',
  clid varchar(80) NOT NULL default "",
  src varchar(80) NOT NULL default "",
  dst varchar(80) NOT NULL default "",
  dcontext varchar(80) NOT NULL default "",
  channel varchar(80) NOT NULL default "",
  dstchannel varchar(80) NOT NULL default "",
  lastapp varchar(80) NOT NULL default "",
  lastdata varchar(80) NOT NULL default "",
  duration int(11) NOT NULL default '0',
  billsec int(11) NOT NULL default '0',
  disposition varchar(45) NOT NULL default "",
  amaflags int(11) NOT NULL default '0',
  accountcode varchar(20) NOT NULL default "",
  uniqueid varchar(32) NOT NULL default "",
  userfield varchar(255) NOT NULL default ""
);
```

### 4.3.2 CONFIGURACIÓN DE ASTERISK PARA USAR CDR

Para almacenar de CDRs en una base de datos mysql se debe configurar el fichero */etc/asterisk/cdr\_mysql.conf*

*[global]*

*hostname=10.1.1.230*

*dbname=CDRsFromMGC*

*table=cdr*

*password=2202621450*

*user=mgc*



Además, es necesario que la generación de CDRs esté activada en el archivo `/etc/asterisk/cdr.conf`:

*[general]*

*; Define whether or not to use CDR logging. Setting this to "no" will override*

*; any loading of backend CDR modules. Default is "yes".*

*enable=yes*

### 4.3.3 EJEMPLO DE UN REGISTRO CDR

Calldate	Clid	Src	Dst	Dcontext
2010-01-15 13:07:5			5121	internal

Channel	Dstchannel
MGCP/aaln/S0/SU2/0@ARMARIO11 -1	MGCP/aaln/S0/SU2/0@ARMARIO12 -1

Lastapp	Lastdata	Duration	Billsec
Dial	MGCP/aaln/S0/SU2/0@ARMARIO12 -1	127	118

Disposition	Amaflags	Accountcode	Userfield	Uniqueid
ANSWERED	3			1263557279.46

Cada uno de los campos almacenados se corresponde con:

- Calldate: comienzo de la llamada. (formato de fecha y hora)
- Clid: caller ID en modo texto (80 caracteres)
- Src: caller ID en número (string, 80 caracteres)
- Dst: extensión de destino (string, 80 caracteres)
- Dcontext: contexto de destino (string, 80 caracteres)
- Channel: Canal usado (80 caracteres)
- DstChannel: Canal de destino (80 caracteres)
- Lastapp: Última aplicación (80 caracteres)
- Lastdata: Datos de la última aplicación (argumentos) (80 caracteres)
- Duration: Tiempo total en el sistema, en segundos (integer)
- Billsec: Tiempo total que la llamada estuvo activa, en segundos (integer)
- Disposition: Estado de la llamada: Contestada (ANSWERED), No contestada (NO ANSWER), Ocupado (BUSY), Fallida (FAILED)
- Amaflags: Toma 4 posibles valores. 1=DEFAULT, 2=OMIT, 3=BILLING y 4=DOCUMENTATION.
- Accountcode: Código de cuenta para facturación de Asterisk (string, 20 caracteres)
- Userfield: Campo de usuario establecido en el plan de llamadas (80 caracteres)
- Uniqueid: Identificador único de canal (32 caracteres)

## **4.4 SERVICIOS DE VALOR AÑADIDO**

### **4.4.1 INTERACTIVE VOICE RESPONSE (IVR)**

El IVR o Respuesta de Voz Interactiva, consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con la persona a través de grabaciones de voz y el reconocimiento de respuestas simples o pulsaciones de números. Está orientado a entregar y/o capturar información por medio del teléfono, permitiendo el acceso a servicios de información u otras operaciones.

#### **4.4.1.1 Ámbitos de uso**

El IVR (Interactive Voice Response) es comúnmente implantado en empresas o entidades que reciben grandes cantidades de llamadas con el objeto de reducir la necesidad de personal y sus costes asociados. Entre otras, podemos mencionar a las bancas telefónicas.

Es habitual utilizar la tecnología de IVR para enrutar una llamada entrante hacia un departamento sin la necesidad de intervención humana, reduciendo así el tiempo de espera de los clientes.

En los centros de atención telefónico al cliente se usan los IVR's para guiar al llamante hacia los agentes con mayor conocimiento de una materia específica.

Además, pueden prestar cualquier clase de servicio: televotación, encuestas, sorteos, acceso a bases de datos, servicios informativos, etc.

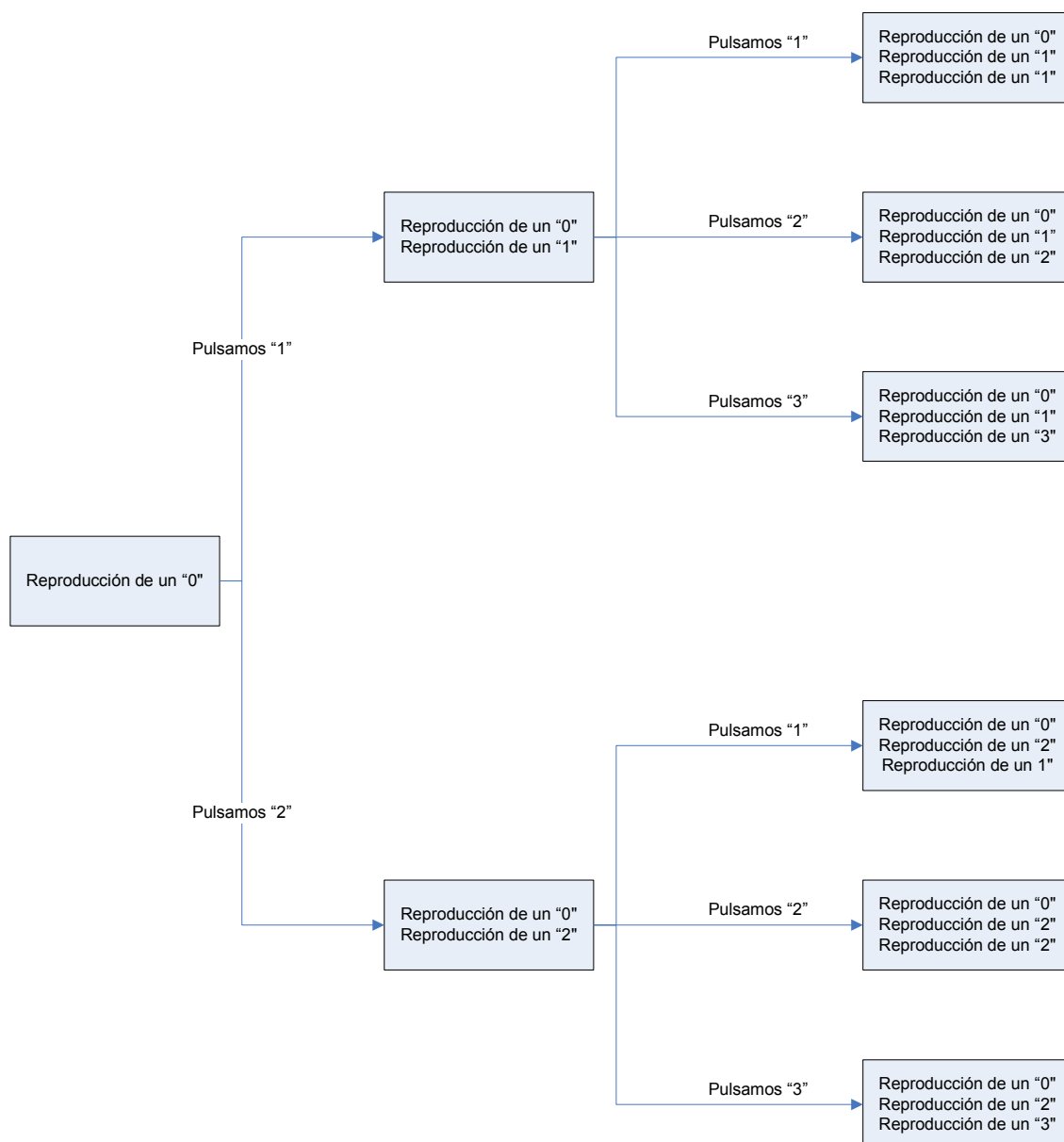
#### **4.4.1.2 ¿Cómo funciona?**

El usuario realiza una llamada a un número de teléfono, el sistema de audiorespuesta contesta la llamada y presenta al usuario una serie de acciones a realizar mediante mensajes (menús de opciones) previamente grabados en archivos de audio (Por ejemplo "Presione uno para ventas, dos para administración"). El usuario selecciona la opción introduciendo un número en el teclado del teléfono y navega por los diferentes menús hasta encontrar la información solicitada o hasta que el sistema enrute la llamada al destinatario elegido.

#### **4.4.1.3 Configuración en Asterisk**

Para obtener un sistema IVR lo más versátil posible es necesario disponer de tecnologías como Text-To-Speech (para convertir texto en audio) y Reconocimiento de Voz. La implantación de dichas tecnologías no ha sido objeto de estudio de este proyecto.

Se ha configurado un menú principal con dos posibles opciones y, cada una de ellas, con varios sub-menús. Las grabaciones que se reproducen corresponden a los números que se pulsán en el teléfono. Es una forma de indicar en qué zona del IVR nos encontramos. Representado gráficamente:



**Ilustración 4.4-1. Esquema de los sub-menús del sistema IVR instalado**

Para implementar esta funcionalidad en el servidor Asterisk es necesario añadir en el archivo *extensions.conf* las siguientes líneas:

```

[ivr_menu]
exten => s,1,Background(en/digits/0); Se reproduce el "0"
exten => s,n,WaitExten(5); Se esperan 5 segundos para que se pulse el siguiente número
exten => 1,1,Goto(ivr_menu_1,s,1); Si se pulsa un "1" se redirige al submenú adecuado
exten => 2,1,Goto(ivr_menu_2,s,1); Si se pulsa un "2" se redirige al submenú adecuado
exten => i,1,Goto(s,1); Si se pulsa una opción inválida, se vuelve al mensaje de inicio
exten => t,1,Goto(s,1); Si se pasa el tiempo de espera, se vuelve al mensaje de inicio
    
```

```
[ivr_menu_1]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,WaitExten(5)
exten => 1,1,Goto(ivr_menu_1_1,s,1)
exten => 2,1,Goto(ivr_menu_1_2,s,1)
exten => 3,1,Goto(ivr_menu_1_3,s,1)
```

```
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_1_1]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,Background(en/digits/1)
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID});Se ejecuta el
;script crear_fichero_dinamico.agi, pasando como argumento el valor de la variable
;UNIQUEID.
```

```
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})
```

```
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_1_2]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,Background(en/digits/2)
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})
```

```
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_1_3]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,Background(en/digits/3)
```

```
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})
```

```
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_2]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Wait(5)
exten => 1,1,Goto(ivr_menu_2_1,s,1)
exten => 2,1,Goto(ivr_menu_2_2,s,1)
exten => 3,1,Goto(ivr_menu_2_3,s,1)
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_2_1]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Background(en/digits/1)
```

```
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-{UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})
```

```
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_2_2]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Background(en/digits/2)
```

```
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-{UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})
```

```
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_2_3]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Background(en/digits/3)
```

```
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-{UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})
```

```
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

Las tabulaciones existentes en el código no son necesarias para el correcto funcionamiento del IVR, pero facilitan su interpretación.

Una característica muy importante de los planes de llamada en Asterisk (y en concreto en los IVRs) es la capacidad de ejecutar scripts externos gracias a AGI (Asterisk Gateway Interface). AGI es una interfaz que permite ejecutar PHP, Perl, Bash, C y Pascal. El único requerimiento es que los ficheros deben tener extensión `.agi`, tener permisos de ejecución y estar situados en el directorio configurado en `/etc/asterisk/asterisk.conf`, por defecto, `astagidir => /var/lib/asterisk/agi-bin`.

También es esencial asignarle una extensión a la que, al llamar, conteste el IVR. En este caso se ha seleccionado el número 5901 añadiendo la siguiente línea al fichero `extensions.conf`:

```
exten => 5901,1,Goto(ivr_menu,s,1)
```

#### 4.4.2 VOICEMAIL

El Voicemail se utiliza para dejar un mensaje si el extremo está comunicando, nadie contesta a la llamada o simplemente se rechaza. En Asterisk la configuración de este servicio se realiza mediante el fichero `/etc/asterisk/voicemail.conf`.

Este archivo contiene un contexto general:

```
[general]
; Enviar archivos en las notificaciones de e-mail
attach=yes
; Usar el formato wav para los mensajes de voz
format=wav
; Limitar el tiempo máximo del mensaje de voz a 180 segundos
maxmessage=180
; Limitar el tiempo mínimo del mensaje a 3 segundos
minmessage=3
; Anunciar el numero que llamó antes de repetir el mensaje
saycid=yes
; Limitar el número de intentos de registro a 3
maxlogins=3
```

En el resto de contextos se definen los buzones de los usuarios. Los usuarios pueden agruparse en contextos.

El formato básico es el siguiente:

[default]

extensión => contraseña, nombre de usuario, email de usuario, email de notificación, opciones

- La extensión hace referencia al número de teléfono llamado.
- La contraseña para acceder al buzón de voz de ese usuario.
- El nombre de usuario es el nombre del cliente de la extensión.
- El email del usuario es el correo al que serán enviados los mensajes.
- El email de notificación es un email alternativo donde pueden ser enviadas las notificaciones para administración o control.
- Las opciones sirven para sobreescribir las del contexto [general] o especificar una zona horaria para el usuario.

Para un correcto funcionamiento del sistema de voicemail, se hace imprescindible contar con un servidor de correo instalado que sea capaz de procesar y enviar los mails correspondientes. En este caso, se ha optado por la instalación del software postfix para la realización de dicha tarea. El detalle de su instalación y configuración se puede encontrar en el correspondiente apéndice.

Para la utilización del voicemail se ha creado una macro que gestiona el estado de la llamada (sin respuesta/ocupado) y redirige, si es necesario, la llamada al buzón del usuario.

[macro-voicemail]

```
exten => s,1,Dial(${ARG1},20)
exten => s,2,Goto(s-$DIALSTATUS,1)
exten => s-NOANSWER,1,VoiceMail(u${MACRO_EXTEN})
exten => s-NOANSWER,2,Hangup()
exten => s-BUSY,1,VoiceMail(b${MACRO_EXTEN})
exten => s-BUSY,2,Hangup()
exten => _s-,1,Goto(s-NOANSWER,1)
```

Esta macro se encuentra incluida en el plan de llamadas (*/etc/asterisk/extensions.conf*) y se utiliza del siguiente modo:

```
exten => 5903,1,Macro(voicemail,SIP/SIP2) ;
```

#### 4.4.3 ACCESO A BASES DE DATOS

Una característica muy importante que dota a Asterisk de gran versatilidad es el acceso a bases de datos. Esta utilidad, unido al uso del IVR, permite un amplio abanico de posibles aplicaciones (consulta de notas, inventarios, realización de pedidos, consulta de facturación, etc....).

Dado el ámbito académico de este proyecto, se va a simular un sistema básico de consulta de calificaciones para los alumnos. Para ello, se crea una base de datos con una tabla llamada `notas_2009`.

```
mysql> create database asignaturas;
mysql> use database asignaturas;
mysql> create table notas_2009 (cod_asignatura int(4) not null, cod_alumno int(6) not null, nota DECIMAL(5,3) not null, key(cod_asignatura));
mysql> insert into notas_2009 VALUES ("1111", "22222", "8");
```

La tabla notas\_2009 se compone de 3 campos, código de asignatura, código de alumno y nota.

Una vez creada la estructura e introducidos los registros de ejemplo, se debe establecer una extensión o contexto que realice las tareas necesarias. Para ello se modifica el fichero `/etc/asterisk/extensions.conf` añadiendo las siguientes líneas:

*;El siguiente contexto permite el acceso a una base de datos creada como prueba.  
;En ella se espera que el usuario pulse un número de 4 cifras, correspondiente al código de la asignatura y un número de 6 dígitos correspondiente al código de alumno. Con ambos datos hace una consulta a la bbdd mysql, devolviendo el valor de su nota.*

```
[asignaturas]
exten => s,1,Read(cod_asignatura,en/digits/1,number,4)
exten => s,n,NoOp(${cod_asignatura})
exten => s,n,Read(cod_alumno,en/digits/2,number,6)
exten => s,n,NoOp(${cod_alumno})
exten => s,n,MYSQL(Connect connid localhost root asdfghjkl asignaturas)
exten => s,n,MYSQL(Query resultid ${connid} SELECT `nota` from `notas_2009` WHERE `cod_asignatura`=${cod_asignatura} AND `cod_alumno`=${cod_alumno})
exten => s,n,MYSQL(Fetch fetchid ${resultid} var1)
exten => s,n,NoOp(${var1})
exten => s,n,MYSQL(Disconnect ${connid})
```

A continuación se detalla cada instrucción:

- `Read(cod_asignatura,en/digits/1,number,4)` → Reproduce la grabación en/digits/1 y guarda la pulsación de 4 números en la variable `cod_asignatura`.
- `NoOp(${cod_asignatura})` → Muestra en la consola de Asterisk el valor de la variable `cod_asignatura`.
- `MYSQL(Connect connid localhost root asdfghjkl asignaturas)` → Crea una conexión con la base de datos `asignaturas` sita en `localhost`, accediendo con usuario `root` y contraseña `asdfghjkl`. El identificador de dicha conexión lo almacena en `connid`.
- `MYSQL(Query resultid ${connid} SELECT `nota` from `notas_2009` WHERE `cod_asignatura`=${cod_asignatura} AND `cod_alumno`=${cod_alumno})` → Query a la base de datos, donde se consulta el campo `nota` de la tabla `notas_2009`, para el código de alumno y de asignatura asignados. El resultado de esta consulta se guarda en `resultid`.
- `MYSQL(Fetch fetchid ${resultid} var1)` → Se extrae el resultado de `resultid` y se guarda en `var1`.
- `MYSQL(Disconnect ${connid})` → Se finaliza la conexión con la base de datos.



Este es el esquema básico de cómo se accede y se interactúa con una base de datos mysql desde Asterisk. Para que la consulta de notas sea completamente operativa, es necesario disponer de un software Text-To-Speech instalado con la finalidad de convertir la calificación en un archivo de audio que el usuario pueda escuchar.

## 4.5 PROBLEMAS DE ASTERISK Y MGCP

El soporte del protocolo MGCP que ofrece Asterisk es muy básico ocasionando diversos problemas a la hora de desarrollar este proyecto. Resumiendo, las limitaciones más importantes han sido:

- Asterisk, en su función como controlador, no solicita que los Media Gateways le notifiquen ciertos eventos, causando que la llamada continuase aunque se haya colgado el teléfono. Su solución consiste en forzar en los routers 1760 la notificación de dichos eventos mediante el comando *persistent on-hook*.
- Asterisk no gestiona correctamente los endpoints de tipo línea analógica ya que no permite marcar números en la línea. Para solucionar este inconveniente se ha implementado un sistema SIP en los media gateways, que, mediante *voice-peers*, redirigira la llamada según se desee.

## 5. MEDIA GATEWAYS

El media gateway es un dispositivo encargado de transformar las tramas de voz/vídeo para las diferentes redes de telecomunicaciones. Entre sus funciones principales destacan la conversión de la señal de audio analógica en paquetes IP, la cancelación de eco y la notificación de eventos (colgado/descolgado del teléfono, pulsación de dígitos, etc.) .

Los Media Gateways habitualmente están controlados por un Media Gateway Controller (en este caso el servidor Asterisk).

### 5.1 HARDWARE EMPLEADO

Para realizar las funciones de Media Gateways se han utilizado los routers Cisco 1760 disponibles en el laboratorio de telemática. En este apartado no se han incluido los teléfonos IP-300 debido a que no cumplen las características esperadas para un Media Gateway (conexión con líneas o teléfonos analógicos).



**Ilustración 5.1-1. Vista trasera y frontal de los routers Cisco 1760 empleados como Media Gateways**

Los routers Cisco disponen de todas las funcionalidades necesarias para, con las tarjetas VIC-FXS y VIC-FXO, comportarse como Media Gateways. Además, la disponibilidad de 14 routers permite el montaje de escenarios complejos y entornos de pruebas que simulen situaciones reales.

Teóricamente, los routers 2 y 3 del laboratorio también se pueden utilizar, con una versión de firmware adecuada, como media gateways. No obstante, esta configuración no ha podido ser comprobada, ya que la versión actualmente instalada en los routers 1721 no dispone del módulo de VoIP.



**Ilustración 5.1-2. Vista delantera y trasera de un router Cisco 1721**

## 5.2 CONFIGURACIÓN

A continuación se detalla el *running-config* correspondiente a uno de los routers 1 del laboratorio, comentándose las secciones relevantes para este proyecto.

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

Es muy importante establecer el hostname de forma adecuada, ya que MGCP lo utiliza como identificador de los endpoints.

```
!
hostname ARMARIO11
!
```

El uso del protocolo SIP requiere que el reloj esté establecido en la zona GMT 0.

```
clock timezone GMT 0
```

Las tarjetas de voz deben estar activadas.

```
!
voice-card 2
!
voice-card 3
!
```

```
no aaa new-model
ip subnet-zero
!
!
!
!
!
!
!
```

Se establece el orden de prioridad de los códecs a utilizar en las llamadas VoIP

```
!
voice class codec 1
  codec preference 1 g726r16
  codec preference 2 g711ulaw
  codec preference 3 g711alaw
  codec preference 4 g728
!
!
!
!
```

Mediante los siguientes comandos se van a configurar los parámetros necesarios para que el router Cisco detecte adecuadamente el tono de colgado enviado por la centralita telefónica Alcatel.

```
!
voice class dualtone-detect-params 1
  freq-max-deviation 25
  freq-max-power 0
  freq-min-power 13
  freq-power-twist 4
  cadence-variation 4
!
!
voice class custom-cptone UAE-CUSTOM-ALCATEL
  dualtone disconnect
  frequency 425
  cadence 375 375
!
!

!
no voice hpi capture buffer
no voice hpi capture destination
!
!
!
```

En la siguiente sección se establecen las políticas de clasificación y marcado de paquetes necesarias para implementar QoS en los switches. En primer lugar, se clasifican los paquetes según su prioridad DSCP y, posteriormente, se marcan con etiquetado CoS en caso de que salgan por una VLAN.

```
!
class-map match-all rtp
  description rtp
  match dscp ef
class-map match-all signaling
  description signaling
  match dscp af31
!
!
policy-map rtp
  class rtp
    set cos 6
  class signaling
    set cos 4
!
!
!
```

La configuración de la interfaz de red permite la conexión con el controlador y el establecimiento de flujos rtp para la realización de llamadas. Se ha creado una subinterfaz en la FastEthernet0/0, con encapsulado 802.1Q y vlan 30. El rango de IPs de esta vlan es 10.4.88.1 - 10.4.88.62, con máscara de red 255.255.255.192. A esta subinterfaz se le aplica la política de marcado de paquetes configurada anteriormente.

```
!
interface FastEthernet0/0
  no ip address
  speed auto
  full-duplex
!
interface FastEthernet0/0.1
  encapsulation dot1Q 30
  ip address 10.4.88.12 255.255.255.192
  service-policy output rtp
```

```
!
interface Serial0/0
  no ip address
  shutdown
!
interface Serial0/1
  no ip address
  shutdown
!
interface ATM1/0
  no ip address
  shutdown
  no atm ilmi-keepalive
  dsl operating-mode auto
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.4.88.1
no ip http server
!
!
!
```

Los puertos de voz 2/X corresponden a la tarjeta FXS a la que se conectan los teléfonos analógicos. El único requerimiento para ellos es que no se encuentren en estado *shutdown*.

```
!
voice-port 2/0
!
voice-port 2/1
```

Por su parte, los puertos de voz 3/X corresponden a la tarjeta FXO.

```
!
voice-port 3/0
  supervisory disconnect dualtone mid-call
  supervisory custom-cptone UAE-CUSTOM-ALCATEL
```

```

supervisory dualtone-detect-params 1
signal groundStart
input gain 14
cptone GB
timeouts call-disconnect 5
timeouts wait-release 5
connection plar opx 5113
caller-id enable
!
!
voice-port 3/1
supervisory disconnect dualtone mid-call
supervisory custom-cptone UAE-CUSTOM-ALCATEL
supervisory dualtone-detect-params 1
signal groundStart
input gain 14
cptone GB
timeouts call-disconnect 5
timeouts wait-release 5
connection plar opx 5113
caller-id enable

```

Es necesario configurar también el apartado correspondiente a MGCP. Se le debe indicar la dirección IP del *Media Gateway Controller*, el codec a utilizar y establecer el modo *persistent onhook* para que el Media Gateway informe al controlador del colgado de un teléfono (esto es debido a una limitación de Asterisk que no solicita el reporte de dicho evento).

```

!
mgcp
mgcp call-agent 10.6.4.6 2427 service-type mgcp version 1.0
mgcp codec g711alaw packetization-period 10
mgcp persistent onhook
!
mgcp profile default
!

```

Los *dial-peer* establecen cómo se va a comportar el Media Gateway con cada llamada de voz. En primer lugar, se configuran los puertos FXS para utilizar MGCP.

```

!
dial-peer voice 1 pots
application mgcpapp
port 2/0
!
dial-peer voice 2 pots
application mgcpapp
port 2/1

```

Las llamadas entrantes con destino 1XX (centralita analógica), las redirige por el puerto 3/0.

```
!  
dial-peer voice 4 pots  
destination-pattern 1..  
direct-inward-dial  
port 3/0  
forward-digits 3
```

Debido a las limitaciones de Asterisk para redirigir una llamada a un endpoint de tipo línea analógica mediante MGCP, se hace indispensable integrar SIP en los routers cisco con el fin de controlar el funcionamiento de las tarjetas FXO.

```
!  
dial-peer voice 6 voip  
destination-pattern T  
session protocol sipv2  
session target ipv4:10.6.4.6:5060  
session transport udp  
dtmf-relay rtp-nte  
codec g711alaw  
clid strip  
no vad  
!  
sip-ua  
retry invite 3  
retry response 3  
retry bye 3  
retry cancel 3  
timers trying 1000  
sip-server ipv4:10.6.4.6  
!  
!  
line con 0  
line aux 0  
line vty 1 4  
login  
!  
no scheduler allocate  
!  
end
```

## 6. TELÉFONOS IP

Los teléfonos IP de los que se dispone, corresponden al modelo IP-300 de la compañía PerfecTone. Estos teléfonos IP pretenden simular, con el firmware adecuado, su funcionamiento como un teléfono analógico y un Media Gateway integrado.



Ilustración 5.2-1. Vista de un teléfono IP-300 disponibles en el laboratorio

Debido a que el firmware por defecto instalado en los teléfonos soporta únicamente el protocolo SIP, fue necesario una actualización del mismo.

### 6.1 FLASH DEL FIRMWARE

Una vez localizada la versión MGCP del firmware para este teléfono, se procedió a su actualización mediante la interfaz web.

El primer paso consistió en hacer click en el botón *upgrade firmware* del interfaz web:

Network Settings					
ip type	static	ppp id		ppp pin	
local ip	10.6.4.10	subnet mask	255.255.255.0	router ip	10.6.4.254
dns	10.1.1.253	dns2	0.0.0.0	mac	00-09-45-51-11-79
Audio Settings					
codec1	g729	codec2	g7231	codec3	gsm
codec4	g711u	codec5	g711a	codec6	null
val	<input checked="" type="checkbox"/>	age	0	aec	<input checked="" type="checkbox"/>
audio frames	2	jitter size	0	g.723.1 high rate	<input checked="" type="checkbox"/>
Phone Settings					
use dialplan	disable	dial number		ddi code	
idd code		idd prefix		ddi prefix	0
inner line	disable	inner line prefix	0	call waiting	<input type="checkbox"/>
forward number	24234880	ftd poweroff	<input type="checkbox"/>	ftd noanswer	<input type="checkbox"/>
ftd always	<input type="checkbox"/>	ftd busy		answer	30
use digllup	<input checked="" type="checkbox"/>	handset in(0-15)	5	handset out(0-31)	9
ring type	dtmf	speaker out(0-31)	15	speaker in(0-15)	8
MGCP Protocol Settings					
service type	huawei	service addr	10.6.4.6:2427	register ttl	60
nat traversal	disable	nat addr		nat ttl	30
phone number	MGCP1P1	endpoint id	end1	domain name	MGCP1P1
register port	2427	rtsp port	1722	rtsp tss	0
local type	normal	dtmf	rfc2833	dtmf payload	101
super password	12345678	debug	output all		
Other Settings					
password	1234	upgrade type	disable	upgrade addr	
snmp ip	0.0.0.0	use daylight	<input type="checkbox"/>		
timezone	(GMT+01:00)Brussels,Copenhagen,Madrid,Paris				
Save Settings		Address Book		Upgrade Firmware	

Ilustración 6.1-1. Screenshot para instalar un nuevo firmware



En la nueva web abierta, se seleccionó la imagen de firmware deseada y se confirmó la operación mediante el botón *Update Firmware*:



**Ilustración 6.1-2. Screenshot para selección de firmware a instalar**



**Ilustración 6.1-3. Screenshot para proceder a la instalación del firmware seleccionado**

La evolución del proceso se muestra en la pantalla del teléfono IP y tras su finalización, se produce un reinicio del terminal IP.

## 6.2 CONFIGURACIÓN

Con la actualización del *firmware* se dispone de opciones MGCP en la interfaz de configuración. Se deben establecer la IP del Media Gateway Controller, el nombre del host y del endpoint en cada uno de los teléfonos.

La configuración se puede realizar a través de la interfaz web o por medio de telnet. Las opciones disponibles son independientes del método de configuración empleado.

Network Settings					
iptype	static	ppp id		ppp pin	
local ip	10.6.4.10	subnet mask	255.255.255.0	router ip	10.6.4.254
dns	10.1.1.253	dns2	0.0.0.0	mac	00-09-45-51-11-79
Audio Settings					
codec1	g729	codec2	g7231	codec3	gsm
codec4	g711u	codec5	g711a	codec6	null
vad	<input checked="" type="checkbox"/>	age	<input type="checkbox"/>	aec	<input checked="" type="checkbox"/>
audio frames	2	jitter size	0	g.723.1 high rate	<input checked="" type="checkbox"/>
Phone Settings					
use dialplan	disable	dial number		ddd code	
idd code		idd prefix		ddd prefix	0
inner line	disable	inner line prefix	0	call waiting	<input type="checkbox"/>
forward number	24234880	fwd poweroff	<input type="checkbox"/>	fwd noanswer	<input type="checkbox"/>
fwd always	<input type="checkbox"/>	fwd busy	<input type="checkbox"/>	answer	30
use digitmap	<input checked="" type="checkbox"/>	handset in(0-15)	5	handset out(0-31)	9
ring type	dtmf	speaker out(0-31)	15	speaker in(0-15)	8
MGCP Protocol Settings					
service type	huawei	service addr	10.6.4.6:2427	register ttl	60
nat traversal	disable	nat addr		nat ttl	30
phone number	MGCP1P1	endpoint id	end1	domain name	MGCP1P1
register port	2427	rtsp port	1722	rtsp tos	0
local type	normal	dtmf	rfc2833	dtmf payload	101
super password	12345678	debug	output all		
Other Settings					
password	1234	upgrade type	disable	upgrade addr	
snmp ip	0.0.0.0	use daylight	<input type="checkbox"/>		
timezone	(GMT+01:00)Brussels,Copenhagen,Madrid,Paris				

Save Settings   Address Book   Upgrade Firmware

Ilustración 6.2-1. Screenshot en el que se muestra la configuración de un teléfono IP-300 con MGCP

La configuración se divide en 5 apartados:

- **Network Settings:** Relativo a configuraciones de red.
- **Audio Settings:** Establecer y priorizar el uso de los códecs para la conversión de audio.
- **Phone Settings:** Configuración relativa a plan de llamadas y características generales del teléfono. Debido a la arquitectura centralizada, el dialplan se almacena únicamente en el controlador.
- **MGCP Protocol Settings:** Esta es la sección relativa a MGCP. Se debe configurar la IP del controlador, el *phone number*, *domain name* y *endpoint id*. También se pueden configurar puertos para el tráfico MGCP y RTP, así como contraseñas y modo de envío de las señales DTMF

La configuración a través de telnet quedaría como sigue:

```
[root@pfc06 ~]# telnet 10.6.4.10
Trying 10.6.4.10...
Connected to 10.6.4.10.
Escape character is '^]'.
SIP-03 V1.42.028 settings
Password:*****
P:\>help
set
get                list settings
store x store current to xth settings
load x  load xth settings to current
exit
write                save settings
P:\>get
*****Network Settings*****
ip 10.6.4.10          subnetmask 255.255.255.0      router 10.6.4.254
dns 10.1.1.253        dns2 0.0.0.0          mac 00-09-45-51-11-79
*****Audio Settings*****
codec1 0[g729]        codec2 1[g7231]      codec3 4[gsm]
codec4 2[g711u]       codec5 3[g711a]      codec6 6[null]
vad 1[enable]         agc 0[disable]       aec 1[enable]
audioframes 2         jittersize 0         6.3k 1[enable]
*****Phone Settings*****
dialplan 0[disable]
innerline 0[disable]  callwaiting 0[disable]
fwdpoweroff 0[disable]
fwdalways 0[disable] fwdbusy 0[disable]   fwdnoanswer 0[disable]
digitmap 1[enable]   handsetin 5          handsetout 9
ringtype 0[dtmf]     speakerout 15        speakerin 8
*****Protocol Settings*****
servicetype 1[huawei]
caaddr 10.6.4.6:2427  registerttl 60
nat traversal 0[disable]
nataddr [empty]      natttl 30
phonenumber MGCP1P1  endpointid end1      domain MGCP1P1
registerport 2427     rtpport 1722         rtptos 0
dtmfpayload 101      localtype 0[normal]  dtmf 1[rfc2833]
*****Other Settings*****
superpassword 12345678 debug 2[output all]
password 1234        upgradetype 0[disable]
upgradeaddr [empty]
sntrip 0.0.0.0        daylight 0[disable]
timezone 28[(GMT+01:00) Brussels, Copenhagen, Madrid, Paris]
test 0 0 0 0 0 0
P:\>
```

**Ilustración 6.2-2. Configuración del teléfono IP-300 mediante telnet**

Este tipo de configuración no ofrece ninguna característica ni opción adicional a la interfaz web. La sintaxis para establecer valores es:

P:\> *set opción valor*

## 6.3 PROBLEMAS

Se encontraron complicaciones en la recepción de llamadas si los teléfonos utilizaban el protocolo MGCP. Durante 2-3 segundos la conversación se realiza correctamente, pero un vez finalizado ese período el teléfono interrumpe el flujo rtp (impidiendo el envío o recepción de audio). En esta situación, se hace necesario el reinicio del terminal IP debido a que se interrumpe el registro con el controlador.

Este es un inconveniente que no ha podido ser solventado. Con el fin de permitir la integración de estos terminales en la maqueta, se realizó su configuración como teléfonos SIP mediante la creación de su correspondiente entrada en el fichero `/etc/asterisk/sip.conf`

## 7. CONFIGURACIÓN DE SOLUCIÓN COMPLEJA

### 7.1 ARMARIOS SIMULANDO DISTINTAS SEDES

Debido a la cantidad de armarios existentes, se requiere una solución que permita su configuración de manera rápida y eficaz.

Con este objetivo, se ha instalado un servidor tftp, en el que se almacenan todos los archivos de configuración de cada armario. Estos ficheros son básicamente iguales, variando únicamente:

- El *hostname*: varía desde ARMARIO1 hasta ARMARIO15.
- La IP: se le asigna  $10.4.88.n^{\circ}_{armario+1}$ , máscara de red 255.255.255.192 y encapsulado 802.1Q en la vlan 30.

Para que los routers puedan acceder al tftp es necesario que dispongan de conectividad con la red del laboratorio de Telemática.

La obtención de la configuración se realiza desde el modo *enable* de los routers cisco:

```
Router#copy tftp: running-config
Address or name of remote host []? 10.6.4.6
Source filename []? armario11.cfg
Destination filename [running-config]?
Accessing tftp://10.6.4.6/armario11.cfg...
Loading armario11 from 10.6.4.6 (via FastEthernet0/0): !
[OK - 1030 bytes]
```

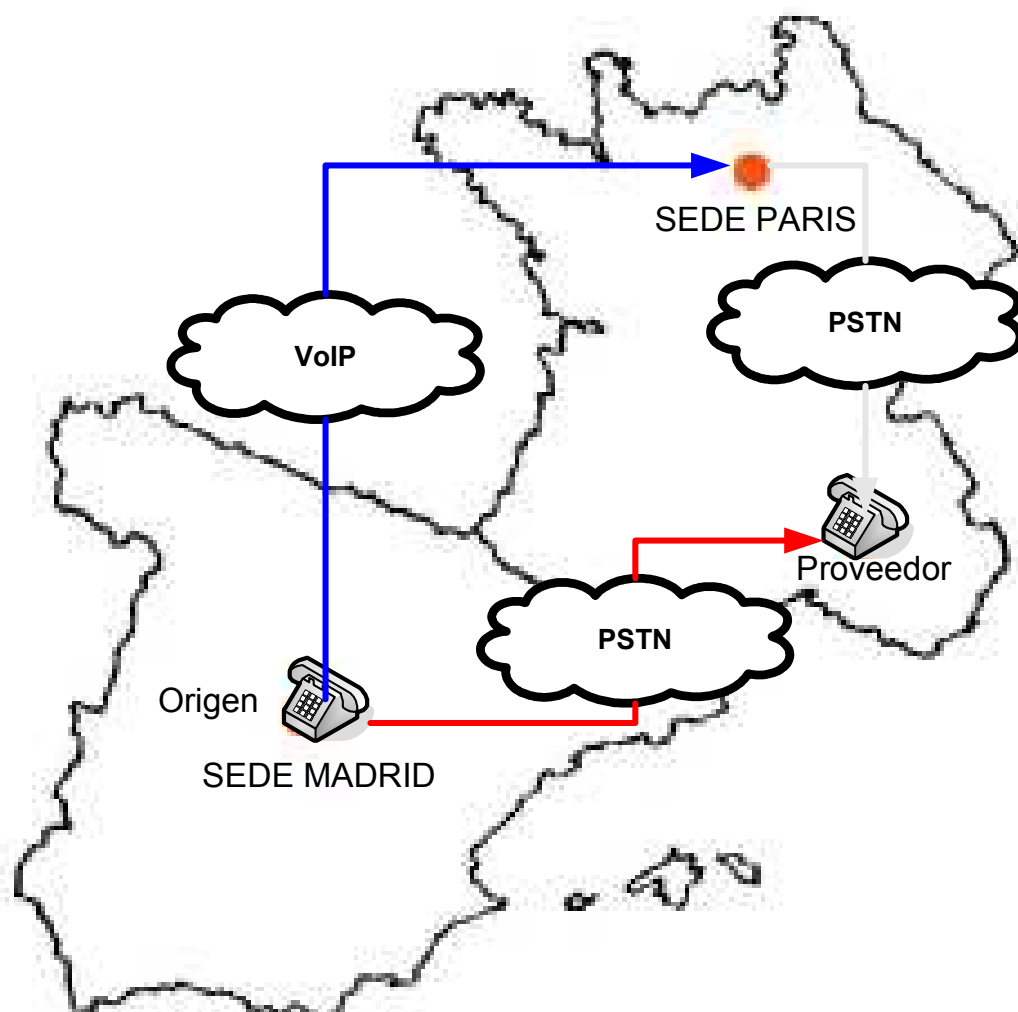
Si se desea establecer esta configuración de modo permanente, se debe copiar el *running-config* al *startup-config*.

El nombre de los ficheros de configuración se ha descrito en función del número de armario, desde *armario1.cfg* a *armario15.cfg*.

Gracias a la infraestructura desplegada, configurar los 15 armarios es una tarea rápida y sencilla.

### 7.2 ENRUTAMIENTO DE LLAMADAS (IP o LÍNEA ANALÓGICA)

Una de las principales características de este sistema es la posibilidad de enrutar la llamada, bien mediante la red IP o la telefonía analógica tradicional. Esta versatilidad permite, entre otras opciones que, en caso de fallo de una de las dos redes sea posible redirigir el tráfico a través de la otra. También es útil para empresas multisede con presencia en distintos países. A modo de ejemplo, supongamos que en una empresa con sedes en Madrid y París, un trabajador de la sede de Madrid desea llamar a un proveedor de Francia. Si se hiciese la llamada normalmente desde España, tendría el coste de una llamada con tarificación europea. En cambio, se podría dirigir la llamada por VoIP hasta la sede de París, enrutando la llamada desde allí por la línea analógica, reduciendo el coste.

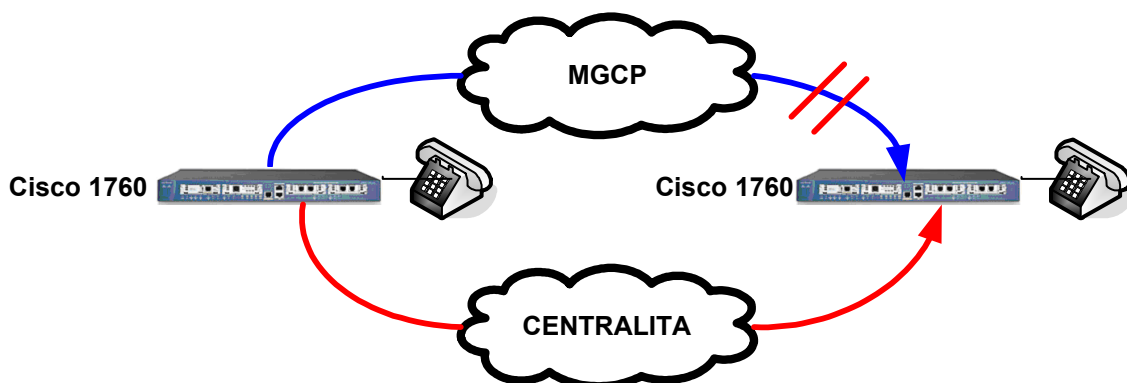


**Ilustración 7.2-1. Realización de llamadas en empresa multisede**

En **rojo** se muestra el camino seguido por la llamada puramente analógica a través de los diferentes operadores que aplican sus tarifas correspondientes.

En **azul** y **gris** se observa la segunda posibilidad para dirigir la llamada. El **azul** correspondería a la parte de VoIP hasta la sede parisina de la empresa y el **gris** indica la terminación de la llamada a su destino a través de la línea analógica francesa.

La maqueta ha quedado configurada de forma que, por defecto, las llamadas se enruten por IP, y en caso de fallo, la llamada se realiza mediante la centralita que interconecta todos los armarios.



**Ilustración 7.2-2. Escenario básico configurado en el proyecto**

La llamada debe ser realizada por el camino mostrado en azul. En caso de que sea imposible alcanzar el destino mediante la red ethernet, la llamada se enrutará a través de la centralita analógica que interconecta los armarios. Con tal finalidad se utilizará la interfaz FXO que llevan incorporada los routers Cisco. El esquema básico es el siguiente:

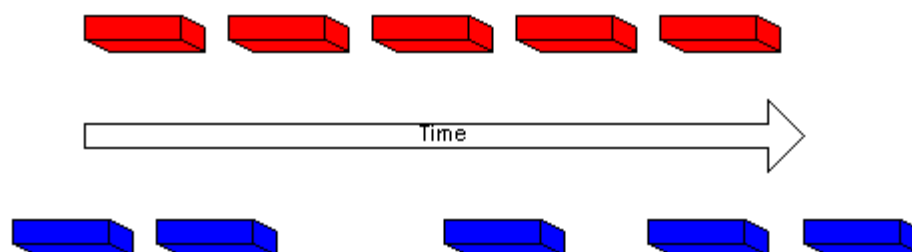
- Un usuario marca el número de teléfono interno con el que desea contactar.
- La comunicación se intenta establecer mediante MGCP.
- En el supuesto de que no sea posible, el Media Gateway origen recibe la orden de llamar, a través de su interfaz FXO, a la extensión analógica asignada al armario de destino.

## 7.3 CALIDAD DE SERVICIO (QOS)

En los sistemas de transmisión de voz en tiempo real existen 3 parámetros que van a determinar la calidad de una llamada:

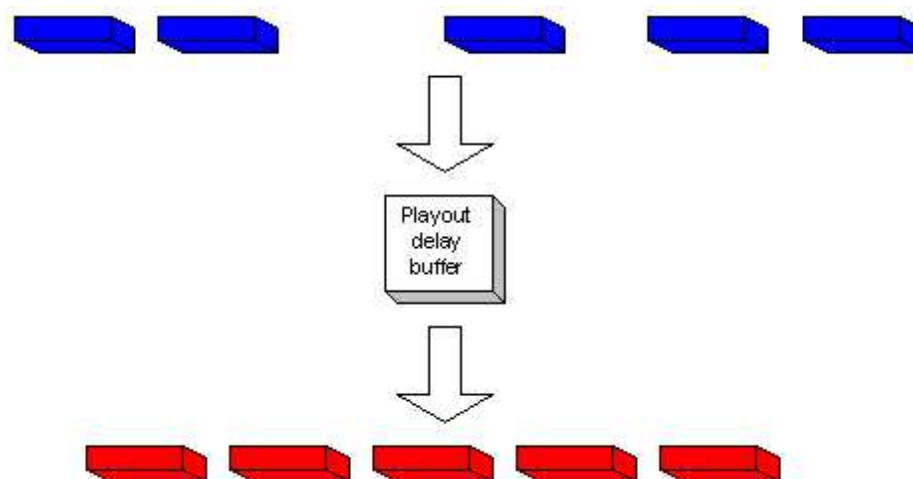
- Pérdida de paquetes: El porcentaje de paquetes que no llegan a su destino mide la pérdida de paquetes de la red que puede producirse por errores en alguno de los equipos o por sobrepasar la capacidad de algún buffer en momentos de congestión. Normalmente, en aplicaciones que no funcionan en tiempo real, este problema se evita con la retransmisión de los paquetes, pero en la telefonía IP esta técnica no puede ser utilizada.
- Latencia: Se define como el retardo entre el envío de paquetes y su recepción. Desde el punto de vista del usuario es el “vacío” en la conversación. La latencia se debe, generalmente, a la red por la que se transmiten los paquetes y al tiempo de procesamiento de los codecs.
- Jitter: Los paquetes del transmisor pueden llegar a su destino con diferentes retardos. Un retardo de un paquete varía impredeciblemente debido a su posición en las colas de los dispositivos de red a lo largo del camino entre el emisor y el destino. Esta variación en retardo se conoce como *jitter* y puede afectar seriamente la calidad del flujo de audio y/o vídeo.

Gráficamente, el jitter se expresa así:



**Ilustración 7.3-1 Jitter introducido en un flujo de datos. En rojo, paquetes enviados. En azul, paquetes recibidos.**

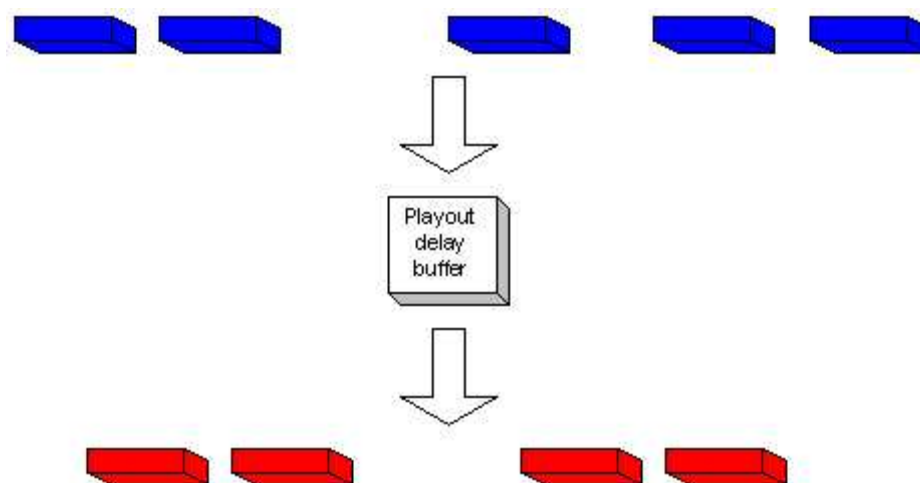
Cuando un router recibe un flujo de audio RTP, debe compensar el jitter que encuentra. El mecanismo que se encarga de esta función es el *playout delay buffer*. Este buffer se encarga de almacenar paquetes y enviarlos, con la separación adecuada entre ellos, al DSP para su procesamiento.



**Ilustración 7.3-2. Funcionamiento del Playout delay Buffer**



Si un paquete llega con excesivo jitter, se descarta.



**Ilustración 7.3-3. Paquete descartado por Jitter excesivo**

Para pérdidas pequeñas como un paquete, el DSP interpola la señal de audio y su ausencia no es perceptible. Cuando el jitter excede el umbral que el DSP puede soportar para reconstruir los paquetes descartados, se perciben distorsiones y ruido.

Con el fin de minimizar la aparición de estos efectos, se hace imprescindible establecer unos límites máximos aceptables para que la llamada se realice con normalidad.

- Pérdida de paquetes: La Voz sobre IP no es tolerante a pérdida de paquetes. Se necesitan pérdidas inferiores al 1% para evitar errores auditivos.
- Latencia: En el documento ITU-T G.114 se recomienda una latencia máxima de 150ms en un sentido.
- Jitter: No hay un consenso acerca del límite aconsejable. Depende de varios factores como el tamaño del buffer de jitter o el retardo existente extremo a extremo. No obstante, para pruebas de laboratorio Cisco recomienda un valor máximo de unos 10ms. En general, un buen valor para establecer el buffer de jitter es equiparlo al retardo existente en el canal.

[6] Cisco Systems. “Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms)”.

[http://www.cisco.com/en/US/tech/tk652/tk698/technologies\\_tech\\_note09186a00800945df.shtml](http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a00800945df.shtml)

### 7.3.1 ESCENARIOS

Es imprescindible determinar los escenarios en los que se va a realizar el estudio de Calidad de Servicio. Se consideró interesante analizar el efecto ocasionado por diferentes variables (distancia geográfica, velocidad de transmisión y llamadas simultáneas). En todos los casos se va a utilizar el codec G.711.

Es interesante realizar un estudio del escenario formado por los armarios del laboratorio, ya que podría servir de base para una futura práctica relacionada con Calidad de Servicio. En esta situación, el retardo de propagación es despreciable debido a que las distancias a recorrer son muy pequeñas. Se dispone de 15 armarios, correspondiendo el número máximo de llamadas a 2 por armario, exactamente las mismas soportadas por el DSP integrado en los routers 1760.

### 7.3.2 ESTUDIO TEÓRICO

El equipamiento de red con el que se va a realizar el estudio consiste en switches catalyst 2950, con 4 colas diferentes para clasificar el tráfico. Además, el códec G.711, tiene las siguientes características: envía 50 paquetes/segundo, separación entre paquetes de 20ms, el BW necesario es 87.2kbps y con un tamaño por paquete de 1.744kbits.

- **Características de las colas del switch:**

Los switches disponen de 4 colas en configuración *weighted round-robin*. El número de paquetes que se escogen en cada ciclo de envío depende de los pesos asignados a cada cola.

Los paquetes se organizan en colas en función del valor de su campo de CoS (Class of Service) descrito en el protocolo 802.1p.

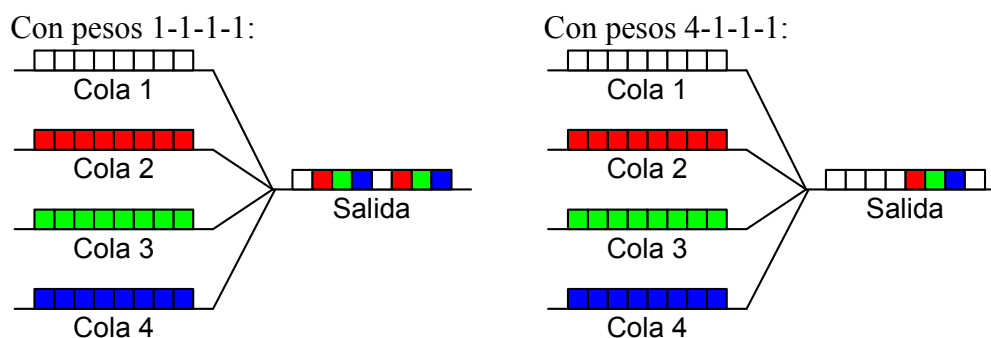


Ilustración 7.3-4. Paquete descartado por Jitter excesivo

- **Influencia de la distancia geográfica:**

Un aumento de la distancia provoca que el retardo de propagación se incremente. Se ha aproximado el retardo total que va a sufrir un paquete:

$$R_{total} = R_{propagación} + R_{transmisión}$$

El retardo máximo de transmisión que se desea permitir es 20ms, correspondiente a la separación entre paquetes consecutivos de una conversación VoIP con el códec G.711.

El retardo total máximo está establecido en 150ms, por lo que no se puede superar el valor de 130ms para el retardo de propagación (salvo si se reduce el retardo de transmisión).

En el siguiente cuadro resumen se presentan los retardos de propagación teóricos y prácticos entre Pamplona y varias ciudades. La distancia estimada es la correspondiente a una línea recta entre ambas ciudades (No es posible conocer la distancia exacta recorrida por los paquetes).

CIUDAD	DISTANCIA(Km)	R <sub>PROPAGACIÓN-TEÓRICO(ms)</sub>	R <sub>PROPAGACIÓN-REAL(ms)</sub>
Moscú	3450	17.25	51.50
New York	5780	28.9	60.85
Toronto	6280	31.4	91.38
Rio Janeiro	7940	39.7	142.15
Sidney	15990	79.95	190.06

**Tabla 7.3-1. Retardos de propagación a diferentes ciudades del mundo**

El retardo teórico se ha calculado según la siguiente ecuación:

$$R_{propagación} = \frac{D(km)}{Velocidad\_propagación(km/ms)} = \frac{D(km)}{200.000.000km/ms}$$

El retardo de propagación real se ha medido utilizando el comando ping dirigido a distintas webs de universidades sitas en las ciudades de destino.

--- *www.msu.ru ping statistics* ---

*13 packets transmitted, 13 received, 0% packet loss, time 12012ms*

*rtt min/avg/max/mdev = 102.035/102.624/103.018/0.367 ms*

--- *www.metropolitan.edu ping statistics* ---

*8 packets transmitted, 8 received, 0% packet loss, time 7007ms*

*rtt min/avg/max/mdev = 121.134/121.379/121.703/0.439 ms*

--- *info.utcc.utoronto.ca ping statistics* ---

*8 packets transmitted, 8 received, 0% packet loss, time 7009ms*

*rtt min/avg/max/mdev = 154.807/158.542/182.757/9.168 ms*

--- *pegasus.udesc.br ping statistics* ---

*16 packets transmitted, 16 received, 0% packet loss, time 15014ms*

*rtt min/avg/max/mdev = 283.601/283.983/284.295/0.661 ms*

--- *solo-rproxy.ucc.usyd.edu.au ping statistics* ---

*22 packets transmitted, 22 received, 0% packet loss, time 21000ms*

*rtt min/avg/max/mdev = 378.589/379.402/380.125/0.867 ms*

Como se desprende de los datos obtenidos, el retardo de propagación correspondiente a Río de Janeiro está ligeramente por encima del máximo recomendado. No obstante, al ser una diferencia pequeña, probablemente la conversación se pueda seguir manteniendo sin problemas notables. El caso de Sydney es más problemático ya que el elevado retardo hace que sea necesario la contratación de líneas especiales de baja latencia para realizar llamadas de VoIP.

- **Influencia de la velocidad:**

Otra característica decisiva en el retardo es la velocidad de transmisión. En este caso se estudiará su influencia durante la realización de una llamada de voz.

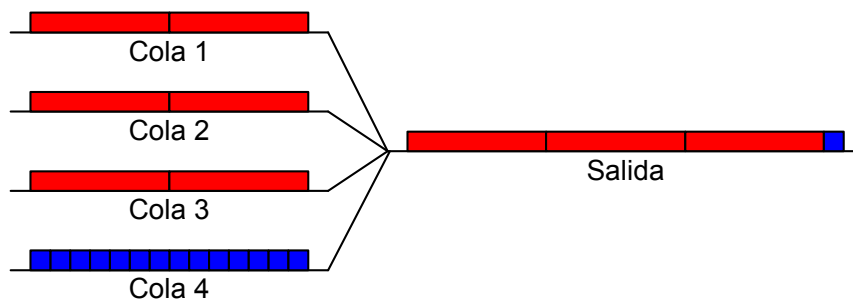
$$R_{tx\_1paquete\_voz} = \frac{1.74kbits}{V_{tx}}$$

Este tráfico de voz se va a clasificar en la cola 4 del switch. El resto de las colas (1, 2 y 3), se rellenan con paquetes de datos de 1500 bytes cada uno (escenario más desfavorable).

$$R_{tx\_1paquete\_datos} = \frac{12000bits}{V_{tx}}$$

Supuesto que únicamente se clasifica el tráfico sin asignar pesos a ninguna de las colas (todas colas con peso 1):

$$R_{envío\_1paquete\_voz\_con\_datos} = 3 \cdot R_{tx\_1paquete\_datos} + R_{tx\_1paquete\_voz}$$



**Ilustración 7.3-5. Tráfico clasificado, sin prioridad**

Se diseña el sistema de tal forma que el retardo de transmisión sea como máximo 20ms, tiempo correspondiente a la separación entre paquetes consecutivos de una conversación de VoIP con el códec G.711.

$$R_{envío\_1paquetes\_voz\_con\_datos} = 20ms = 3 \cdot R_{tx\_1paquete\_datos} + R_{tx\_1paquete\_voz}$$

$$20ms = 3 \cdot \frac{12000bits}{V_{tx}} + \frac{1413bits}{V_{tx}}$$

$$V_{tx} = \frac{36000bits + 1413bits}{0.02s} = 1.87Mbits/s$$

A efectos de retardo, para soportar una llamada de voz clasificando únicamente el tráfico, es suficiente con que la velocidad de transmisión sea superior a 1.87Mbits/s.

Si el enlace tiene una capacidad inferior a 1.87Mbits/s, se debería estudiar otras estrategias, como reducir el tamaño de los paquetes de datos o el número de colas habilitadas. El peso no puede ser superior al número de llamadas ya que afectaría negativamente al retardo y al jitter (en estas circunstancias, algún paquete de voz se procesaría en un intervalo posterior a los 20 ms que le corresponde). En la siguiente tabla se muestra el tiempo de transmisión de un paquete de voz y un paquete de datos en función del ancho de banda del enlace.

Vtx	Rtx_voz	Rtx_datos
256kbits/s	5.39ms	46.88ms
512kbits/s	2.70ms	23.44ms
1024kbits/s	1.35ms	11.72ms
1.87Mbits/s	0.74ms	6.42ms

Tabla 7.3-2. Retardo de transmisión en función de la velocidad de transmisión

Como se puede observar, a bajas velocidades la transmisión de un paquete de datos ya supone más tiempo que los 20ms. En este caso se hace imprescindible fragmentar el paquete en trozos más pequeños o limitar la MTU del enlace.

Si se establecen pesos superiores al peso de llamada (Vtx 1024Kbits/s):

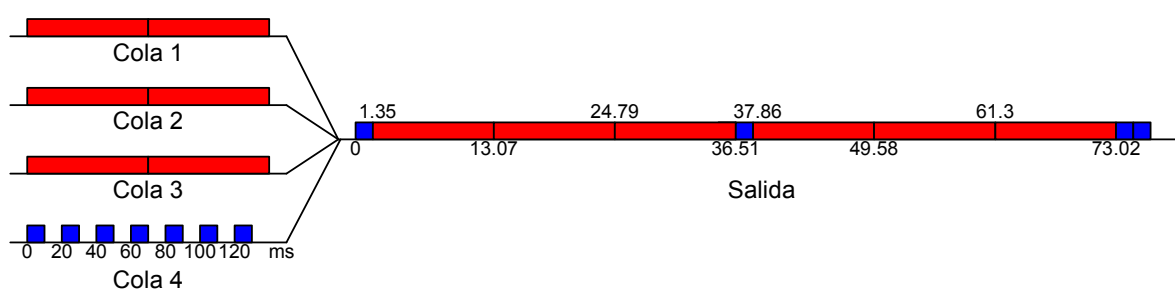


Ilustración 7.3-6. Problemas inherentes con pesos superiores al número de llamadas

Los tres primeros paquetes de VoIP llegan con retardos similares pero el tercero y cuarto se transmiten de forma consecutiva, con lo que el jitter crecerá considerablemente. Además, el tercer paquete (correspondiente al milisegundo 40), se envía a partir del 73.02, superando el parámetro de diseño deseado.

- Influencia del número de llamadas:**

Otro parámetro crítico es la cantidad de llamadas de voz simultáneas que se desean poder realizar. Para este estudio se ha establecido la velocidad de transmisión en 10Mbits/s (velocidad que será utilizada en los tests del laboratorio).

El máximo retardo de transmisión admisible sigue siendo, al igual que en el caso anterior, de 20 milisegundos.

Retardo de transmisión de un paquete de voz:

$$R_{tx\_1paquete} = \frac{1.74kbits}{10Mbits/s} = 0.174ms$$

Retardo de transmisión de un paquete de datos de 1500 bytes:

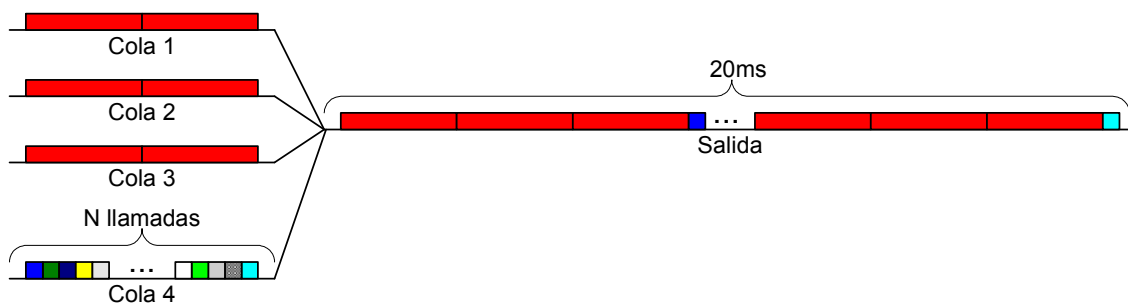
$$R_{tx\_1paquete\_datos} = \frac{12000bits}{10Mbits/s} = 1.2ms$$

Retardo de envío de un paquete de voz y 3 de datos:

$$R_{envio\_1paquete\_voz\_con\_datos} = 3 \cdot R_{tx\_1paquete\_datos} + R_{tx\_1paquete\_voz} + R_{cola} = 3.774ms$$

Retardo de envío de N paquetes de voz y 3 de datos:

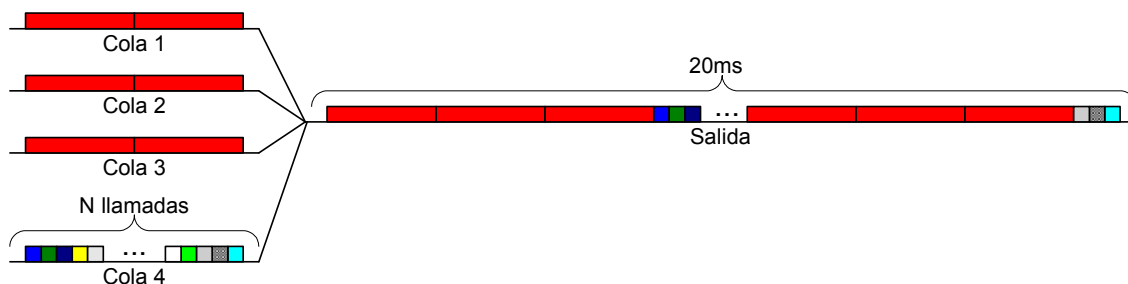
Si no se establecen pesos:



**Ilustración 7.3-7. Colas sin prioridad y varias llamadas simultáneas**

Para pocas llamadas se podrá cumplir el objetivo. No obstante, si se desea un elevado número de llamadas simultáneas, será necesario aumentar el peso de la cola 4 correspondiente a los paquetes de VoIP:

Para un caso con pesos 1-1-1-3:



**Ilustración 7.3-8. Colas con peso 1-1-1-3**

Generalizando para un caso de pesos 1-1-1-X y extrayendo su función analítica:

$$R_{envío\_N\_paquetes\_voz\_con\_datos} \leq 20ms \geq \frac{N}{X} \cdot (3 \cdot R_{tx\_1paquete\_datos} + X \cdot R_{tx\_1paquete\_voz})$$

N	Peso necesario
1	0.18 → 1
5	0.94 → 1
10	1.97 → 2
20	4.36 → 5
30	7.31 → 8
40	11.06 → 12
50	15.96 → 16
75	39.02 → 40
80	47.62 → 48
90	75.28 → 76
91	79.33 → 80
92	83.74 → 84
93	88.55 → 89
94	93.83 → 94
95	99.65 → 100

**Tabla 7.3-3. Pesos necesarios en función del número de llamadas en el sistema**

El máximo número de llamadas que, teóricamente, se pueden establecer de manera simultánea son 94. Para 95 llamadas se requeriría un peso de 100, superior al número de llamadas. Según se ha expuesto con anterioridad esta situación implica que algunos paquetes de una llamada se envíen en los 20ms posteriores a su rango, siendo perjudicial para el retardo y el jitter.

- **Escenario del laboratorio:**

En este caso, el retardo de propagación es despreciable. La velocidad de transmisión de la red se va a establecer en 10Mbps/s.

Suponiendo que las 30 llamadas de voz que se desean se van a iniciar en el mismo instante (escenario más desfavorable):

$$R_{tx\_1paquete} = \frac{1.744kbits}{10Mbps/s} = 0.174ms$$

$$R_{envío\_30paquetes} = 30 \cdot R_{tx\_1paquete} = 4.14ms$$

Este tráfico de voz se va a clasificar en la cola 4 del switch. El resto de las colas (1,2 y 3), se rellenan con paquetes de datos de 1500 bytes cada uno (escenario más desfavorable).

$$R_{tx\_1paquete\_datos} = \frac{12000bits}{10Mbps/s} = 1.2ms$$

Supuesto que se realice únicamente clasificación del tráfico, sin asignar pesos a ninguna de las colas (todas colas con peso 1):

$$R_{\text{envío\_1 paquete\_voz\_con\_datos}} = 3 \cdot R_{\text{tx\_1 paquete\_datos}} + R_{\text{tx\_1 paquete\_voz}} = 3.774ms$$

$$R_{\text{envío\_30 paquetes\_voz\_con\_datos}} = 30 \cdot (3 \cdot R_{\text{tx\_1 paquete\_datos}} + R_{\text{tx\_1 paquete\_voz}}) = 113.22ms$$

La separación entre paquetes consecutivos de una conversación de VoIP con el códec G.711 es de únicamente 20ms. Teóricamente, va a costar 113.22ms enviar el primer paquete de voz de cada una de las 30 llamadas, por lo que se producirá acumulación de los paquetes de voz. Se debe diseñar el sistema de tal forma que envíe los 30 paquetes en un tiempo inferior a 20ms.

$$R_{\text{envío\_30 paquetes\_voz\_con\_datos}} \leq 20ms \geq \frac{30}{X} \cdot (3 \cdot R_{\text{tx\_1 paquete\_datos}} + X \cdot R_{\text{tx\_1 paquete\_voz}})$$

Siendo X el peso que debemos establecer en la cola destinada al tráfico VoIP.

Resolvemos la ecuación y obtenemos un valor de  $X=7.31$ , por lo que el mapa de pesos de la cola 4 del switch Catalyst debe ser:

<b>COLA</b>	1	2	3	4
<b>PESO</b>	1	1	1	8

### 7.3.3 ESTUDIO PRÁCTICO EN LABORATORIO

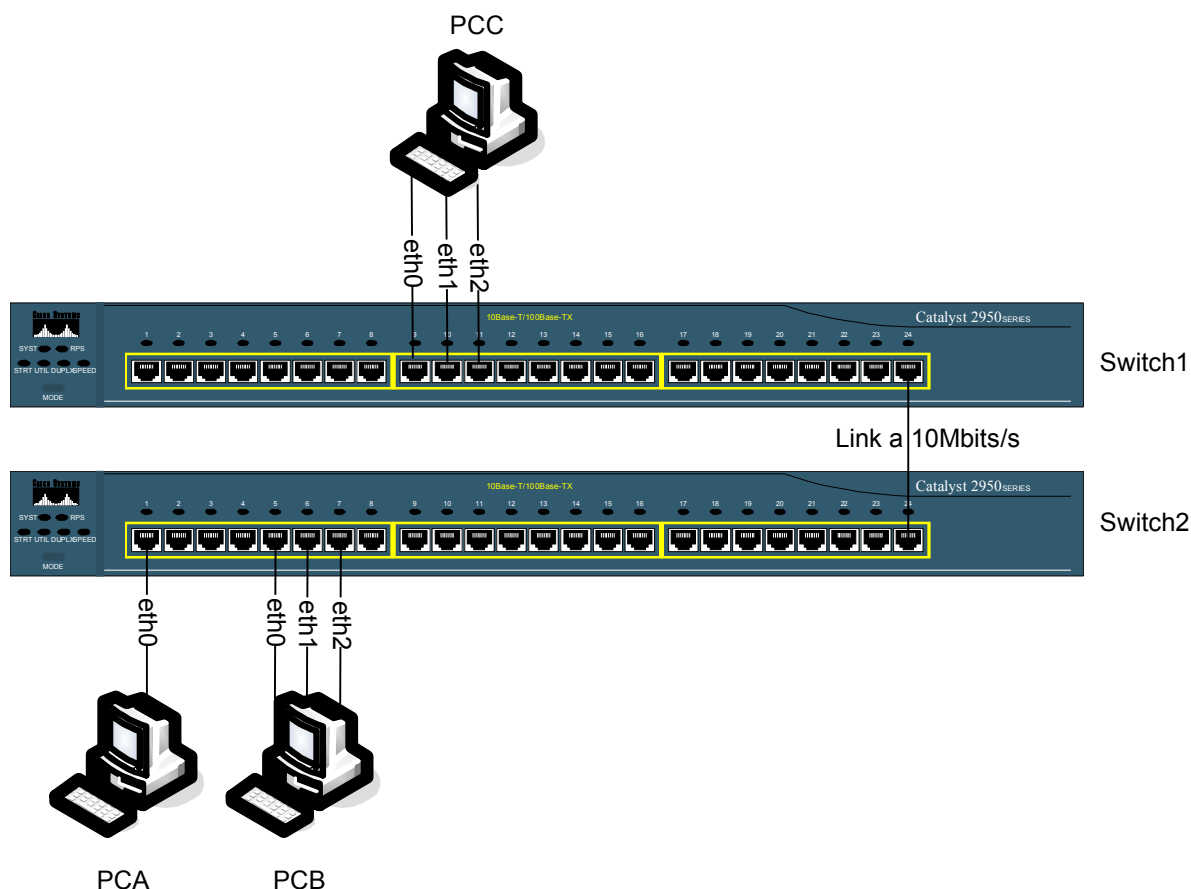
Una vez realizado el estudio teórico, se hace imprescindible comprobar que el diseño funciona correctamente en un entorno de test real. Se definen las siguientes características de diseño:

- 30 llamadas simultáneas.
- Códec G.711.
- Velocidad de transmisión de 10 Mbits/s.

Para su realización, se van a utilizar el PCA, el PCB, el PCC, el Switch1 y el Switch2 de uno de los armarios. El tráfico necesario para la simulación se va a generar mediante el software D-ITG (<http://www.grid.unina.it/software/ITG/>), desarrollado por el "Dipartimento di Informatica e Sistemistica" de la "Universita' degli Studi di Napoli Federico II". Este software no interviene en la sincronización de los relojes de los distintos hosts. Por lo tanto, a la hora de realizar mediciones *one-way delay time*, se requiere del uso de un protocolo como *ntp* o de otras fuentes externas para la sincronización. El generador de tráfico también ofrece la solución de analizar el retardo utilizando *round-trip delay time* en cuyo caso, no es necesario sincronismo entre los relojes.



El esquema básico de montaje es el siguiente:



**Ilustración 7.3-9. Escenario para la comprobación del funcionamiento de las políticas de QoS**

Se necesitan 4 interfaces de red para enviar tráfico a las 4 colas disponibles en el switch. La redirección del tráfico se realiza estableciendo un valor CoS que servirá para clasificarlo.

Los valores CoS que se asignan a cada uno de los puertos del switch se corresponden a diferentes colas según la siguiente tabla:

Valor CoS	0	1	2	3	4	5	6	7
Cola de Prioridad	1	1	2	2	3	3	4	4

El PCA es el encargado de generar el tráfico RTP correspondiente a las 30 llamadas VoIP. La interfaz eth0 se conecta al puerto 1 del switch2. El tráfico de este puerto se clasifica con prioridad CoS 6, saliendo por la cola 4.

El PCB, por su parte, va a inundar el resto de colas del switch con paquetes de 1500 bytes.

La interfaz fastethernet fa0/24 de ambos switches se va a utilizar como un *uplink* de 10Mbps/s entre ambos.

El PCC es el destinatario de todo el tráfico generado.

En la siguiente tabla resumen se observan las IPs configuradas en cada PC.

INTERFAZ	PCA			PCB			PCC		
	IP	Switch	Cola	IP	Switch	Cola	IP	Switch	Cola
ETH0	10.3.19.1	Fa0/1 S1	4	10.3.19.2	Fa0/5 S1	1	10.3.19.3	Fa0/9 S2	X
ETH1	X	X	X	192.168.0.2	Fa0/6 S1	2	192.168.0.3	Fa0/10 S2	X
ETH2	X	X	X	192.168.1.2	Fa0/7 S1	3	192.168.1.3	Fa0/11 S2	X
ETH3	192.168.2.1	Ovislink	X	192.168.1.2	Ovislink	X	192.168.2.3	Ovislink	X

**Tabla 7.3-4. Tabla resumen de las IPs establecidas en cada interfaz**

La interfaz eth3 de los PCs se utiliza para sincronizar sus relojes mediante *ntp* permitiendo realizar mediciones *one-way*. Para asegurarnos de que su precisión es suficiente, se realizará una medición con *RTT* para comparar los resultados.

D-ITG requiere de la creación de ficheros de configuración con los datos acerca del tráfico que se desea generar:

- Tráfico VoIP:

Se genera en el PCA y se compone de 30 llamadas de 5 minutos con paquetes generados cada 20ms.

En el script se especifica con 30 líneas como esta:

```
-a 10.3.19.3 -rp 90XX -t 300000 VoIP -x G.711.2
```

Modificando XX por el número de llamada correspondiente.

La sintaxis significa que se genere tráfico hacia la IP 10.3.19.3, con puerto remoto de destino 90XX, una duración de 5 minutos (300000 milisegundos), de tipo VoIP y códec G.711.

- Tráfico de datos:

```
-a 10.3.19.3 -rp 9101 -t 240000 -d 60000 -T UDP -C 417 -c 1458 (5Mbits/s)
```

```
-a 10.3.19.3 -rp 9102 -t 180000 -d 120000 -T UDP -C 167 -c 1458 (2Mbits/s)
```

```
-a 10.3.19.3 -rp 9105 -t 60000 -d 240000 -T UDP -C 417 -c 1458 (5Mbits/s)
```

```
-a 192.168.0.3 -rp 9103 -t 120000 -d 180000 -T UDP -C 834 -c 1458 (10Mbits/s)
```

```
-a 192.168.1.3 -rp 9104 -t 120000 -d 180000 -T UDP -C 834 -c 1458 (10Mbits/s)
```

Lo más destacable de cada una de las trazas de tráfico es el ancho de banda que utilizan y en qué segundo comienzan (opción *-d*). También es interesante observar que hasta el minuto 3 no se rellenan las colas 2 y 3 del switch.

Una vez que generado el tráfico, es necesario configurar los switches para que realicen las funciones de QoS. Como se va a medir *one-way delay*, no se requieren políticas de QoS en el switch1.

**Configuración:**

```

version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch
!
Definimos los pesos asignados a cada cola.
!
wrr-queue bandwidth 1 1 1 8
ip subnet-zero
!
vtp mode transparent
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
!
!

```

Se crea la vlan 5

```

!
vlan 5

```

Configuración del interfaz por el que va la Voz sobre IP. Se le establece la vlan 5 y un CoS con valor 6.

```

!
interface FastEthernet0/1
 switchport access vlan 5
 switchport trunk native vlan 5
 switchport mode access
 mls qos cos 6
 mls qos cos override

```

Las 3 siguientes interfaces corresponde al tráfico de datos. Se encuentran asignadas a la vlan 5 y con diferentes valores CoS.

```

!
interface FastEthernet0/5
 switchport access vlan 5
 switchport mode access
!
interface FastEthernet0/6
 switchport access vlan 5
 switchport mode access
 mls qos cos 2
 mls qos cos override
!
interface FastEthernet0/7

```

```

switchport access vlan 5
switchport mode access
mls qos cos 4
mls qos cos override

```

La interfaz 24 es la de uplink entre ambos switches y se encuentra limitada a una velocidad de 10Mbps/s. Este enlace, además, se configura en modo *trunk*.

```

!
interface FastEthernet0/24
switchport trunk native vlan 5
switchport mode trunk
speed 10
!
interface Vlan1
no ip address
no ip route-cache
shutdown
!
ip http server
!
line con 0
line vty 5 15
!
!
end

```

La configuración muestra unos pesos asignados a las colas de 1-1-1-8. En el entorno de test se han variado dichos pesos con el objetivo de analizar diferentes casos.

Con el fin de realizar un primer estudio con similares características al cálculo teórico, se han deshabilitado diferentes protocolos del escenario:

- ARP en los PCs
- Cisco Discovery Protocol, Vlan Trunk Protocol y el *keep-alive* en los switches.

Estos protocolos generan paquetes de pequeño tamaño que mejoran las prestaciones

En este caso, los pesos que se han estudiado son:

- Pesos 1-1-1-8
- Pesos 1-1-1-9

El análisis correspondiente a las tramas del tráfico de datos no se ha incluido en la presente memoria. Se puede acceder a los resultados mediante los ficheros de log creados en formato electrónico:

```

ITGDec casol_send_trafico_constant_eth0_log_txt -t
ITGDec casol_send_trafico_constant_eth1_log_txt -t
ITGDec casol_send_trafico_constant_eth2_log_txt -t

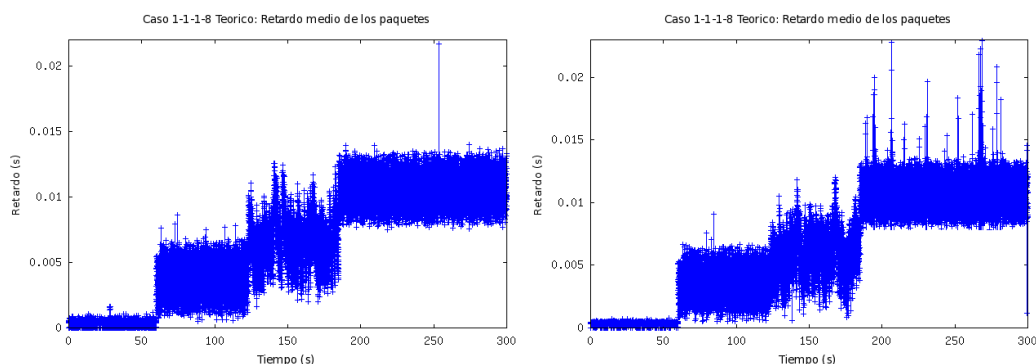
```

De esta forma se mostrará un análisis similar al caso de VoIP, pero con las trazas de datos. Las gráficas también se pueden extraer de las tramas del tráfico de datos.

## **PESOS 1-1-1-8 CASO TEÓRICO**

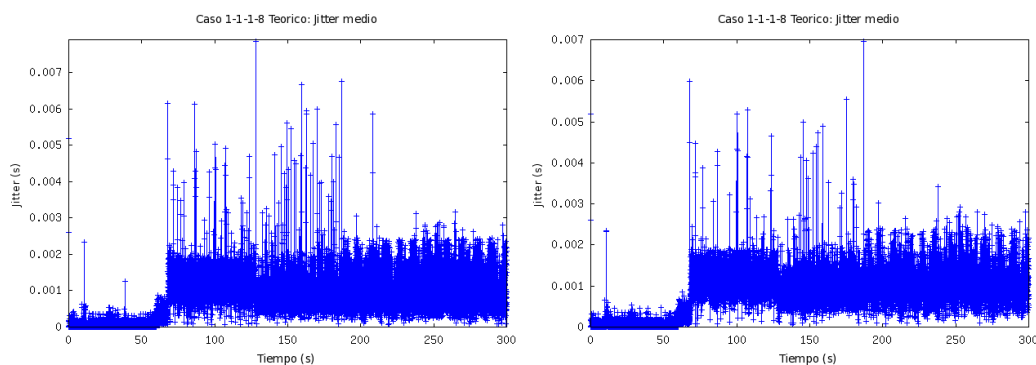
	<b>1 trama VoIP</b>	<b>Media de las 30 llamadas VoIP</b>
<i>Total packets</i>	15000	450000
<i>Minimum delay</i>	0.000470 s	0.000458 s
<i>Maximum delay</i>	0.021698 s	0.022438 s
<i>Average delay</i>	0.007241 s	0.007155 s
<i>Average jitter</i>	0.000620 s	0.000621 s
<i>Delay standard deviation</i>	0.003157 s	0.003115 s
<i>Bytes received</i>	2580000	77400000
<i>Average bitrate</i>	68.805098 Kbit/s	2060.356363 Kbit/s
<i>Average packet rate</i>	50.003705 pkt/s	1497.352008 pkt/s
<i>Packets dropped</i>	0 (0.00 %)	0 (0.00 %)

En primer lugar se presentan las gráficas correspondientes al retardo para una única llamada y para la media de todas llamadas. Cabe destacar que, a partir del tercer minuto (cuando todas las colas están saturadas), varios paquetes superan el umbral establecido de 20ms.



**Ilustración 7.3-10. Retardo caso teórico pesos 1-1-1-8**

Respecto al jitter, los gráficos se encuentran situados por debajo de los 10ms recomendados por Cisco, por lo que en este aspecto el sistema funciona correctamente.



**Ilustración 7.3-11. Jitter caso teórico pesos 1-1-1-8**

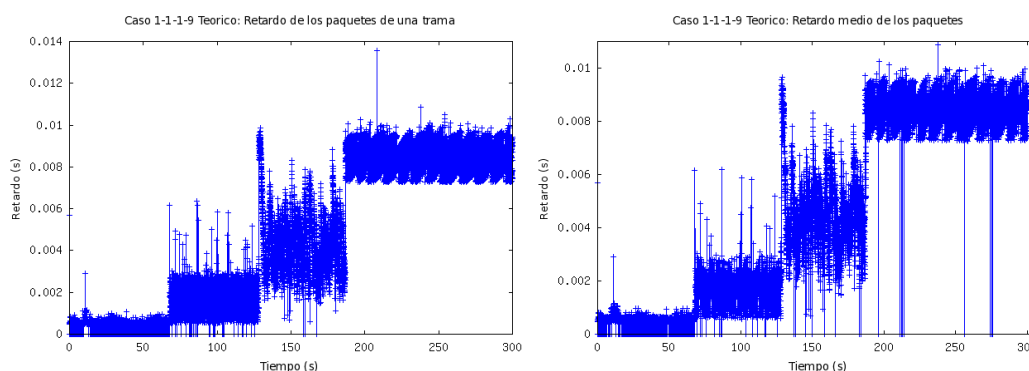
Tal como era de esperar los resultados de este escenario no son tan optimistas como reflejaban los cálculos teóricos. Las causas principales son el uso de dos switches, el retardo de procesamiento de los paquetes tanto en los PCs como en los switches y la velocidad de transmisión.

### **PESOS 1-1-1-9 CASO TEÓRICO**

En la búsqueda de un sistema que cumpla las condiciones de diseño de retardo, jitter y pérdida de paquetes, se aumenta el peso de la cola de VoIP respecto a lo calculado teóricamente.

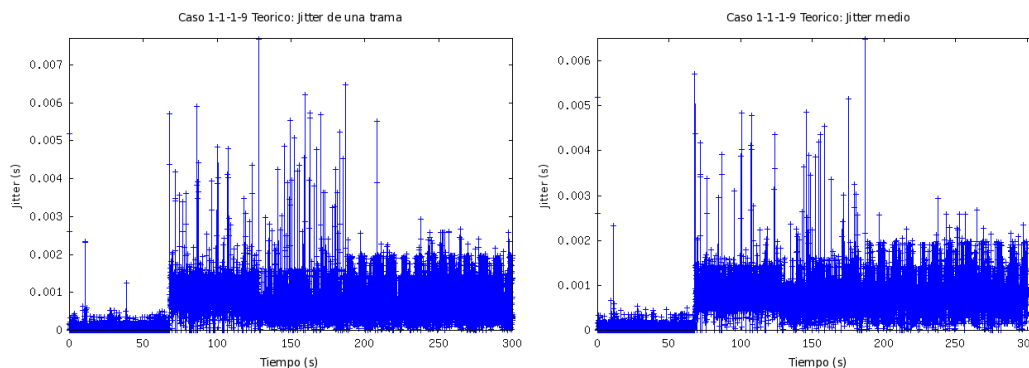
	<b>1 trama VoIP</b>	<b>Media de las 30 llamadas VoIP</b>
<i>Total packets</i>	15000	450000
<i>Minimum delay</i>	0.000449 s	0.000447 s
<i>Maximum delay</i>	0.013565 s	0.019030 s
<i>Average delay</i>	0.004519 s	0.004650 s
<i>Average jitter</i>	0.000652 s	0.000638 s
<i>Delay standard deviation</i>	0.003432 s	0.003446 s
<i>Bytes received</i>	2580000	77400000
<i>Average bitrate</i>	68.802528 Kbit/s	2060.924908 Kbit/s
<i>Average packet rate</i>	50.001837 pkt/s	1497.765195 pkt/s
<i>Packets dropped</i>	0 (0.00 %)	0 (0.00 %)

En este caso ningún paquete sobrepasa los 20ms de retardo, como muestran las siguientes gráficas.



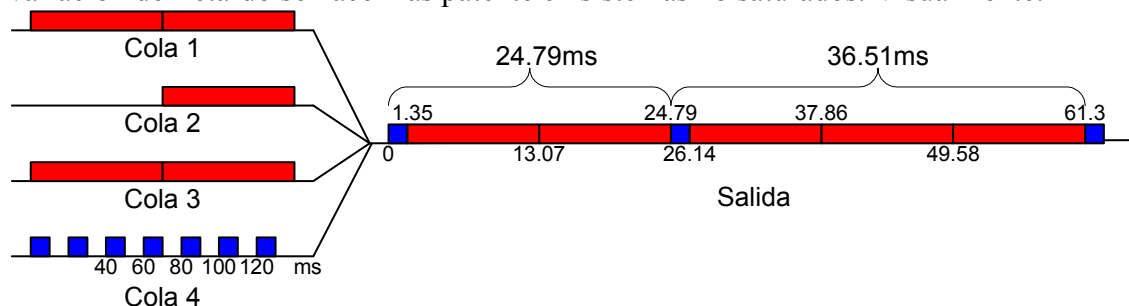
**Ilustración 7.3-12. Retardo caso teórico pesos 1-1-1-9**

Se pueden observar las 4 zonas diferenciadas dependiendo del tráfico de datos introducido en el sistema.



**Ilustración 7.3-13. Jitter caso teórico pesos 1-1-1-9**

Respecto al jitter, el peor comportamiento se presenta cuando el switch no está completamente saturado de tráfico. El resultado es acorde a lo esperado ya que la variación del retardo se hace más patente en sistemas no saturados. Visualmente:



**Ilustración 7.3-14. Efecto de colas no saturadas en el jitter**

Entre el primer y el segundo paquete de voz la diferencia en tiempo es de 24.79ms, mientras que entre el segundo y el tercero, asciende a 36.51ms. Esto es debido a que en el segundo tramo, se transmite un paquete adicional de datos, por lo que a efectos de jitter, es preferible una situación saturada

Las siguientes simulaciones corresponden a un entorno más real, sin desactivar las características mencionadas anteriormente. Con esta nueva situación, se espera obtener mejores resultados debido a la existencia en el sistema paquetes pequeños como pueden ser de ARP o del CDP. Estos paquetes, al poseer un tamaño menor tienen un retardo de transmisión inferior, permitiendo que los paquetes de VoIP puedan ser procesados con mayor celeridad.

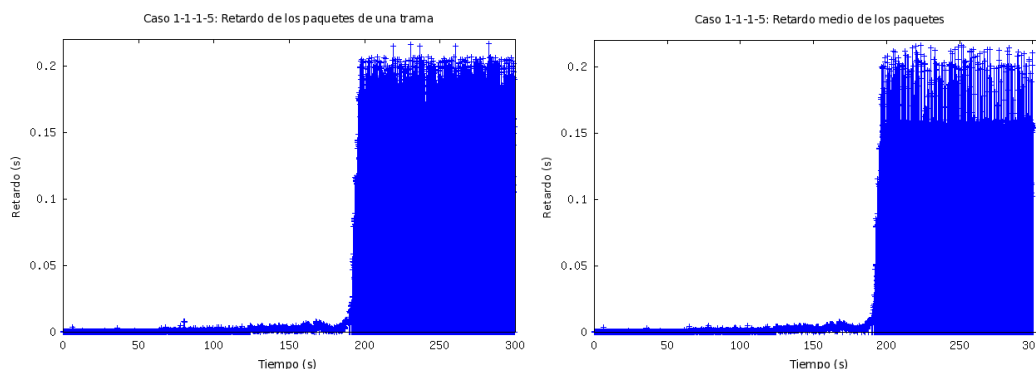
**PESOS 1-1-1-5**

Con esta configuración de pesos, obtenemos los siguientes resultados para una llamada:

	<b>1 trama VoIP</b>	<b>Media de las 30 llamadas VoIP</b>
<i>Total packets</i>	14900	445227
<i>Minimum delay</i>	0.000298 s	0.000290 s
<i>Maximum delay</i>	0.216764 s	0.219447 s
<i>Average delay</i>	0.054280 s	0.054408 s
<i>Average jitter</i>	0.009351 s	0.009550 s
<i>Delay standard deviation</i>	0.074360 s	0.074553 s
<i>Bytes received</i>	2562800	76579044
<i>Average bitrate</i>	68.504168 Kbit/s	2041.244081 Kbit/s
<i>Average packet rate</i>	49.785006 pkt/s	1483.462268 pkt/s
<i>Packets dropped</i>	58 (0.39 %)	3728 (0.83 %)

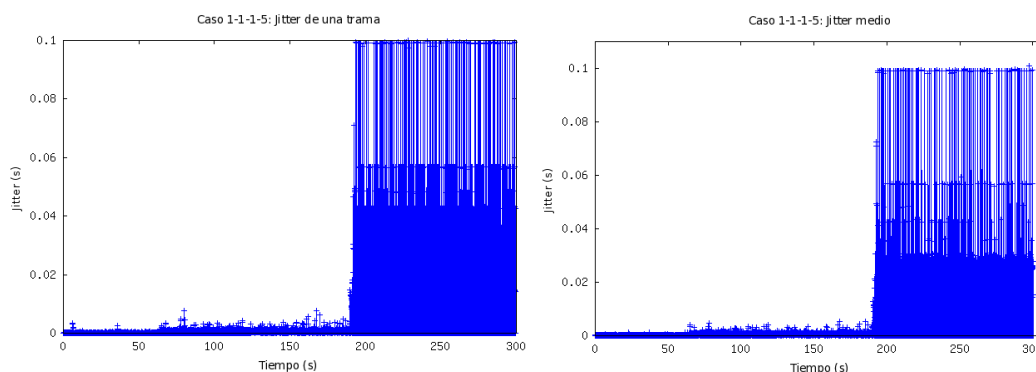
Como se puede observar, el retardo máximo es muy elevado y la cantidad de paquetes perdidos, aunque baja, es inaceptable para una correcta comunicación.

Las siguiente gráficas representan el retardo, jitter y pérdida de paquetes para ambos casos.



**Ilustración 7.3-15. Retardo pesos 1-1-1-5**

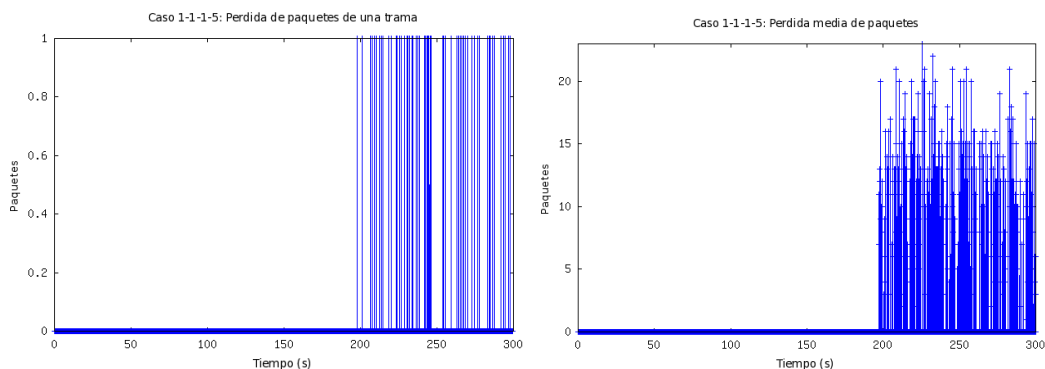
El retardo aumenta considerablemente a partir del segundo 180-190 que es cuando se introducen 20Mbits/s de tráfico de datos y las colas del switch se llenan.



**Ilustración 7.3-16. Jitter pesos 1-1-1-5**



El jitter también aumenta, situándose en torno a los 0.1 segundos al final del test. Este valor es superior al recomendado por Cisco para las pruebas de laboratorio. En ocasiones, se puede realizar un intercambio entre jitter y retardo, aumentando o disminuyendo el tamaño del buffer de jitter. En este caso, ambos parámetros se encuentran por encima de los límites máximos establecidos, por lo que no es posible el intercambio.

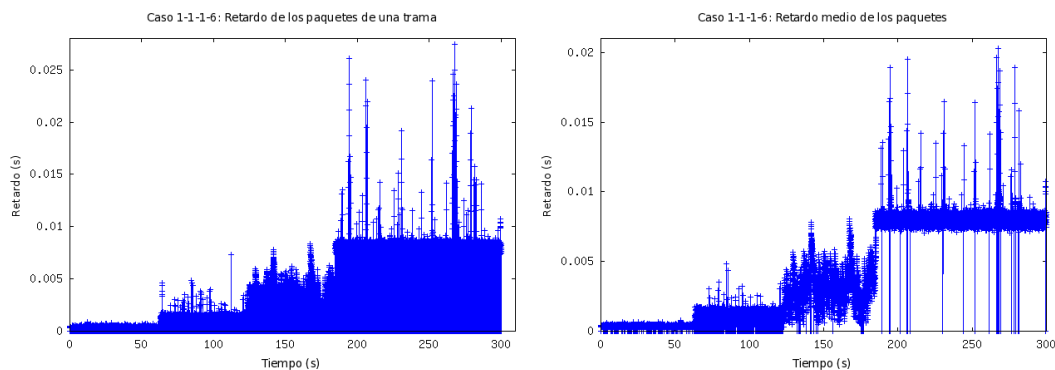


**Ilustración 7.3-17. Pérdida de paquetes pesos 1-1-1-5**

Se observa que existen pérdidas de paquetes a partir del segundo 200 aproximadamente. Estas pérdidas son inaceptables en un entorno de transmisión en tiempo real, afectando simultáneamente a más de 20 de las llamadas realizadas.

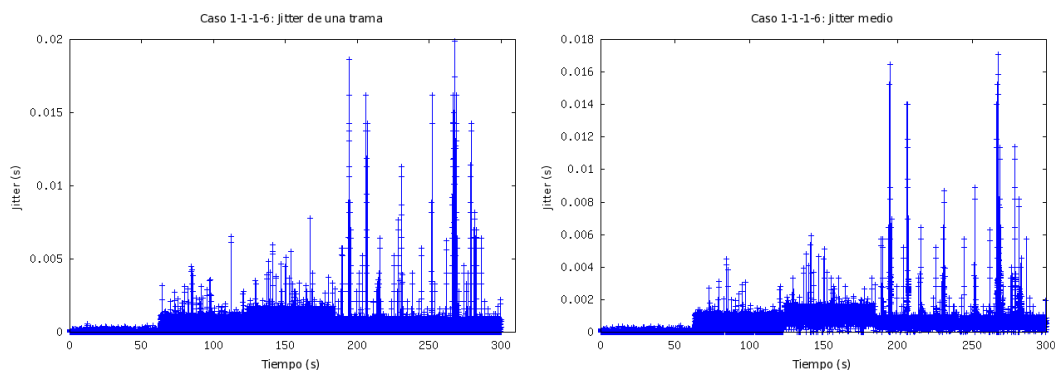
### **PESOS 1-1-1-6**

	<b>1 trama VoIP</b>	<b>Media de las 30 llamadas VoIP</b>
<i>Total packets</i>	14950	448650
<i>Minimum delay</i>	0.000292 s	0.000291 s
<i>Maximum delay</i>	0.027225 s	0.031045 s
<i>Average delay</i>	0.003982 s	0.003985 s
<i>Average jitter</i>	0.000642 s	0.000587 s
<i>Delay standard deviation</i>	0.003490 s	0.003495 s
<i>Bytes received</i>	2571400	77167800
<i>Average bitrate</i>	68.802766 Kbit/s	2057.762599 Kbit/s
<i>Average packet rate</i>	50.002010 pkt/s	1495.467005 pkt/s
<i>Packets dropped</i>	0 (0.00 %)-	0 (0.00 %)



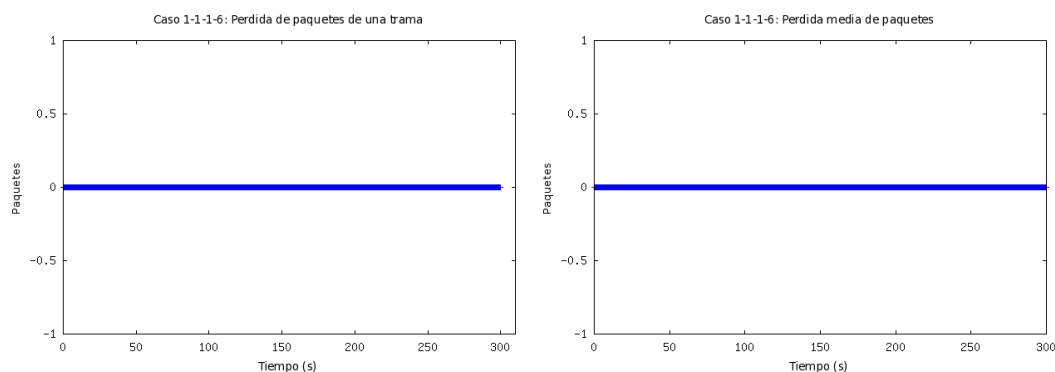
**Ilustración 7.3-18. Retardo pesos 1-1-1-6**

El retardo en este caso se mantiene por debajo del límite recomendado de 150ms. No obstante, en algunos paquetes es superior a los 20ms estimados en el diseño teórico.



**Ilustración 7.3-19. Jitter pesos 1-1-1-6**

El jitter es superior a los 10ms recomendados, por lo que ésta solución sigue sin ser plenamente válida. En esta situación, el jitter tiene su peor comportamiento en saturación. Observando el retardo, en saturación muestra unos picos muy elevados provocando un incremento en el jitter. Los picos de retardo por encima de 20ms, significan que hay paquetes de voz que se procesan en el intervalo posterior al deseado.



**Ilustración 7.3-20. Pérdida de paquetes pesos 1-1-1-6**

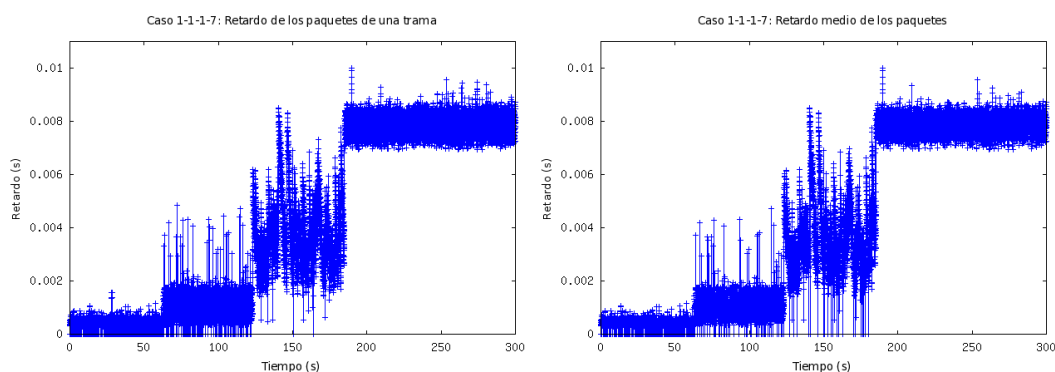
En resumen, el retardo máximo se encuentra en un valor superior a los 20ms establecidos pero se encuentra por debajo de 150ms. El jitter también supera el valor recomendado por Cisco, no obstante, se podría aumentar el tamaño del buffer de jitter a costa de un incremento en el retardo.

### **PESOS 1-1-1-7**

	<b>1 trama VoIP</b>	<b>Media de las 30 llamadas VoIP</b>
<i>Total packets</i>	15000	448600
<i>Minimum delay</i>	0.000295 s	0.000289 s
<i>Maximum delay</i>	0.010007 s	0.013981 s
<i>Average delay</i>	0.004092 s	0.003972 s
<i>Average jitter</i>	0.000492 s	0.000484 s
<i>Delay standard deviation</i>	0.003247 s	0.003357 s
<i>Bytes received</i>	2580000	77159200
<i>Average bitrate</i>	68.802876 Kbit/s	2057.529032 Kbit/s
<i>Average packet rate</i>	50.002090 pkt/s	1495.297262 pkt/s
<i>Packets dropped</i>	0 (0.00 %)	0 (0.00 %)

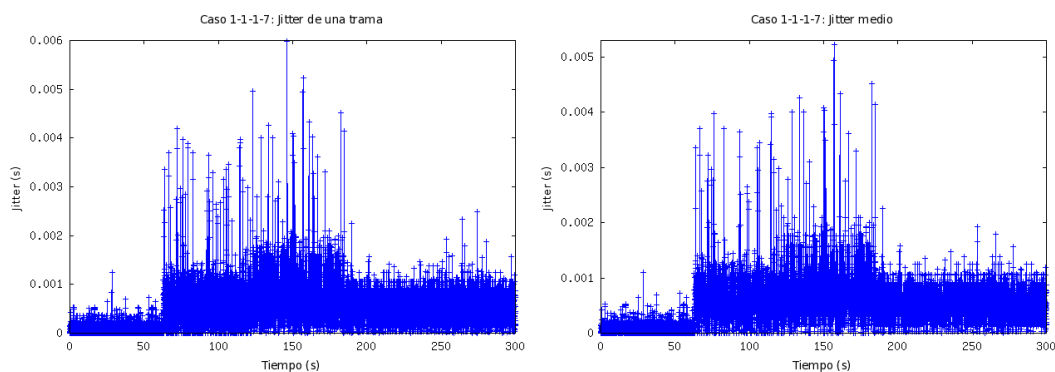
Este escenario cumple con los requerimientos establecidos: 0% de pérdida de paquetes y un retardo máximo de 13ms. Es más bajo que el teórico calculado (20ms) debido a la existencia de otro tipo de tráfico no generado por D-ITG y al hecho de que no comiencen las 30 llamadas en el mismo milisegundo exacto.

Observando las gráficas de retardo y jitter:



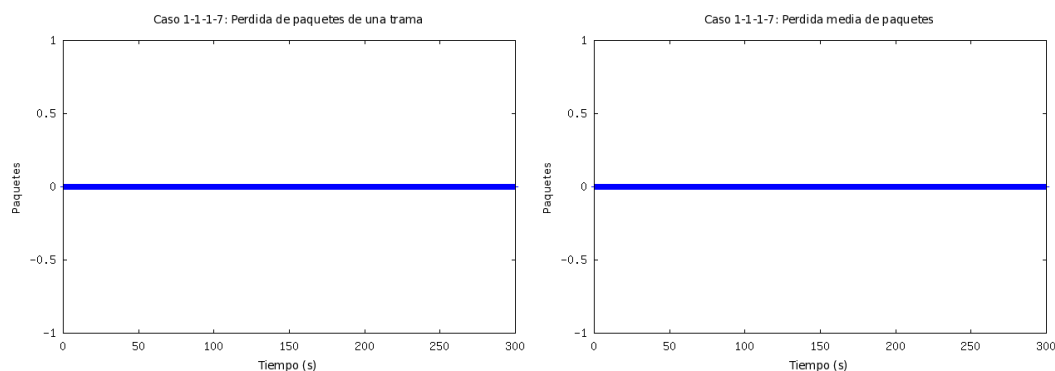
**Ilustración 7.3-21. Retardo pesos 1-1-1-7**

El retardo aumenta conforme se introduce tráfico en el switch2, pero sin superar los 14ms de máximo.



**Ilustración 7.3-22. Jitter pesos 1-1-1-7**

Analizando las gráficas correspondientes al jitter, éste no supera el umbral de 10ms recomendado por Cisco.



**Ilustración 7.3-23. Pérdida de paquetes pesos 1-1-1-7**

Las pérdidas de paquetes son inexistentes, al igual que en el caso anterior.

Tras analizar los resultados y gráficas, se concluye que el caso 1-1-1-7 tiene las características deseadas y permite la realización de las 30 llamadas de voz, independientemente del tráfico que soporte el switch2 (siempre y cuando no se inunde la cola 4 con tráfico ajeno a la VoIP).

### **PESOS 1-1-1-7 RTT**

En este caso se ha utilizado la técnica de medir el retardo de ida y vuelta en vez de usar el protocolo NTP.

	<b>1 trama VoIP</b>	<b>Media de las 30 llamadas VoIP</b>
<i>Total packets</i>	15000	450000
<i>Minimum delay</i>	0.000291 s	0.000290 s
<i>Maximum delay</i>	0.009230 s	0.015704 s
<i>Average delay</i>	0.004131 s	0.004132 s
<i>Average jitter</i>	0.000475 s	0.000502 s
<i>Delay standard deviation</i>	0.003353 s	0.003359 s
<i>Bytes received</i>	2580000	77400000
<i>Average bitrate</i>	68.802744 Kbit/s	2061.516608 Kbit/s
<i>Average packet rate</i>	50.001994 pkt/s	1498.195209 pkt/s
<i>Packets dropped</i>	0 (0.00 %)	0 (0.00 %)

Los resultados son similares a los obtenidos usando NTP, comprobando que la precisión utilizada ha sido correcta.

Puesto que las gráficas de resultados de este caso son prácticamente iguales al apartado anterior, no se han incluido en la presente memoria.

### **7.3.4 LIMITACIÓN DEL ANCHO DE BANDA**

Es importante dar prioridad al tráfico marcado como VoIP, pero también es necesario limitar el impacto que pueda tener en el resto de las comunicaciones. En concreto, los armarios de prácticas están interconectados con el resto de la infraestructura de red del Departamento de Automática y Computación, por lo que un exceso de tráfico en la cola 4, podría influir negativamente en el rendimiento de la red.

Por ello, en los switches que sirvan como conexión entre los armarios del Laboratorio y el resto de la red del Departamento, se debería establecer una política similar a la configuración siguiente testada en un switch 2950 con *Enhanced Image*.

```
Current configuration : 1865 bytes
```

```
!
```

```
version 12.1
```

```
no service pad
```

```
service timestamps debug uptime
```

```
service timestamps log uptime
```

```
no service password-encryption
```

```
!
```

```
hostname Switch
```

```
!
```

```
!
```

Se crean las clases necesarias para marcar el tráfico de señalización y rtp que se genera debido a las llamadas de VoIP.

```
!
class-map match-all rtp
  description rtp
  match ip dscp 46
class-map match-all signaling
  description signaling
  match ip dscp 26
!
```

Una vez marcados los paquetes se configura la política para limitar el ancho de banda. Se establece un límite de 5Mbps/s para cada una de las clases con la acción predeterminada de tirar los paquetes que sobrepasen dicho límite.

```
!
policy-map BW_limitation
  class rtp
    police 5000000 8192 exceed-action drop
  class signaling
    police 5000000 8192 exceed-action drop
!
ip subnet-zero
!
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
```

Finalmente se aplica la política para limitar el ancho de banda a las interfaces necesarias.

```
!
interface FastEthernet0/1
  no ip address
  service-policy input BW_limitation
!
interface FastEthernet0/2
  no ip address
!
interface FastEthernet0/3
  no ip address
!
interface FastEthernet0/4

!
interface Vlan1
  no ip address
  no ip route-cache
  shutdown
!
ip http server
!
```

```

!
line con 0
line vty 0 4
  login
line vty 5 15
  login
!
end

```

Con esta configuración se consigue que el tráfico prioritario no supere los 5Mbits/s. Si se satura el canal, se observa que la velocidad media se mantiene en 4845Kbits/s (más cabeceras) y el exceso de tráfico es descartado por el switch:

	Ancho de banda limitado
<i>Total packets</i>	12467
<i>Bytes received</i>	18176886
<i>Average bitrate</i>	<b>4845.352593 Kbit/s</b>
<i>Average packet rate</i>	415.410888 pkt/s
<i>Packets dropped</i>	<b>137523 (91.69 %)</b>

## 7.4 CONFIGURACIÓN EN ALTA DISPONIBILIDAD

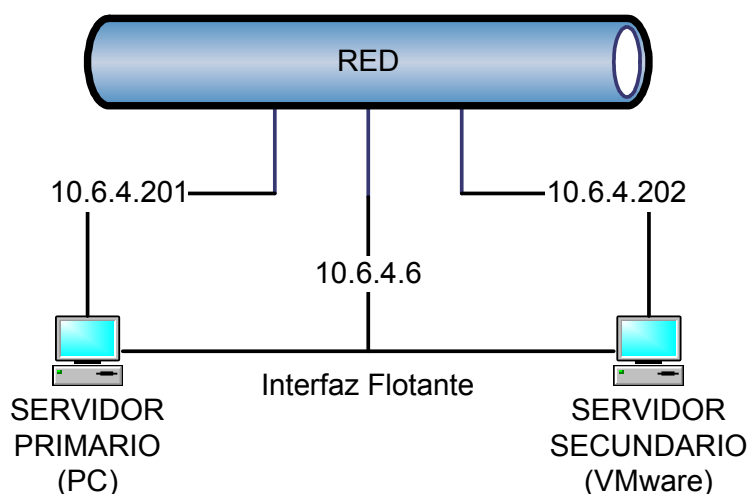
Las comunicaciones son una parte muy importante de cualquier empresa, gobierno o entidad. Por ello, se hace imprescindible asegurar, en la medida de lo posible, un correcto funcionamiento de las mismas.

El primer componente que ofrece alta disponibilidad consiste en la existencia de 2 posibilidades de enrutamiento de las llamadas, mediante VoIP o analógico, tal y como se ha comentado en el apartado 6.2.

No obstante, es necesario asegurar un correcto funcionamiento del *Media Gateway Controller*. Se ha configurado un segundo pc (máquina virtual) como servidor Asterisk secundario. El software no incluye características para su utilización en cluster, y este tipo de soluciones se dejan en manos de herramientas de terceros (*heartbeat, sun cluster, symantec cluster, etc.*).

Implementar un sistema de estas características es costoso, por lo que se ha optado por una configuración más sencilla que ofrezca alta disponibilidad a la solución aunque sin llegar al nivel de un cluster.

El diseño es el siguiente:



**Ilustración 7.4-1. Esquema de red del Asterisk maestro y esclavo**

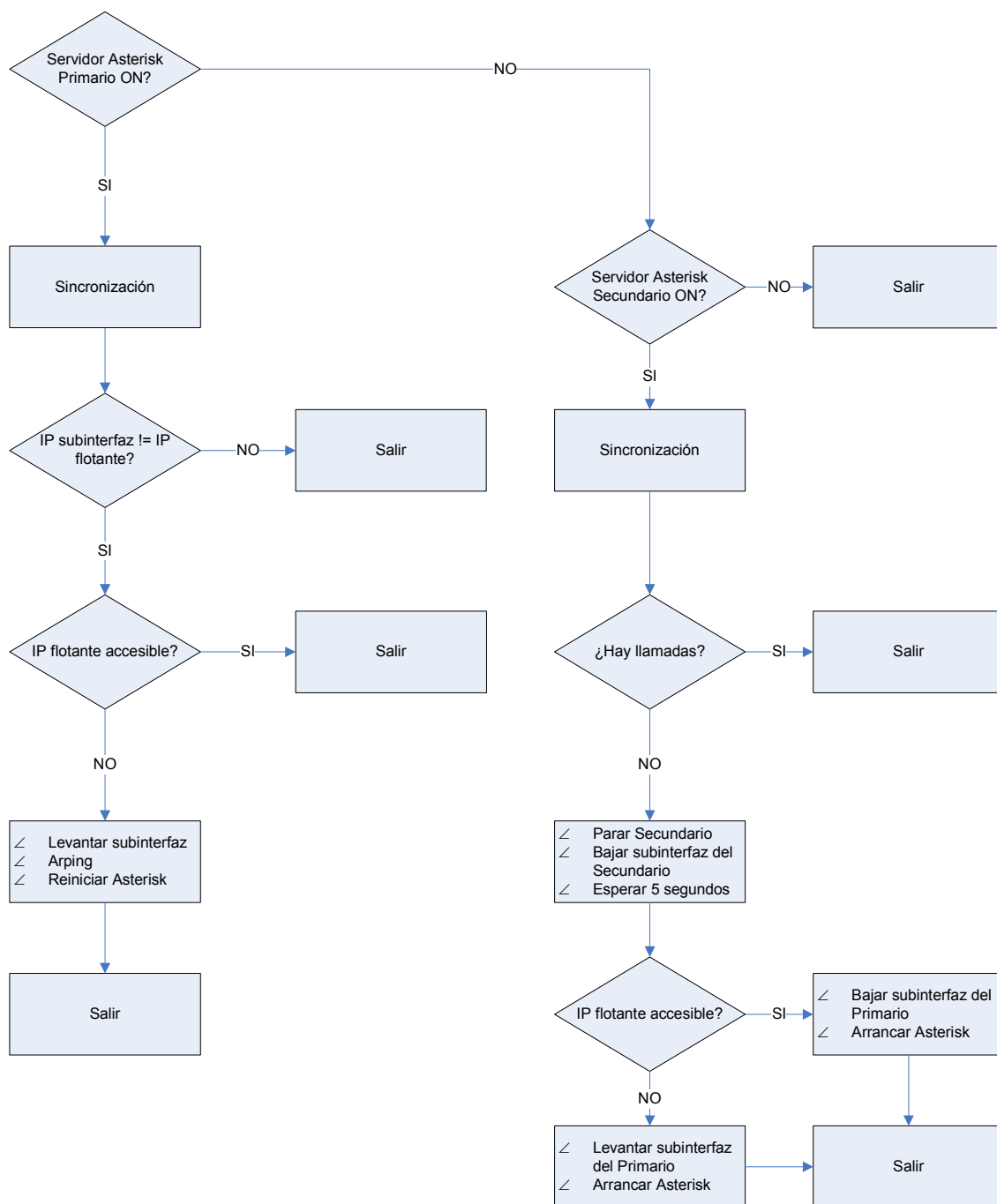
Cada nodo tiene un interfaz real conectado a la red de telemática, y comparten un subinterfaz flotante con la IP que ofrece el servicio de *Media Gateway Controller* (10.6.4.6).

El funcionamiento se basa en scripts que corren en el servidor primario y secundario. La sincronización de archivos (para que ambos servidores posean la misma información) se realiza mediante la herramienta rsync.

Los scripts se encuentran adjuntos en formato electrónico y en el apartado correspondiente del anexo.



- ESQUEMA SCRIPT NODO PRIMARIO

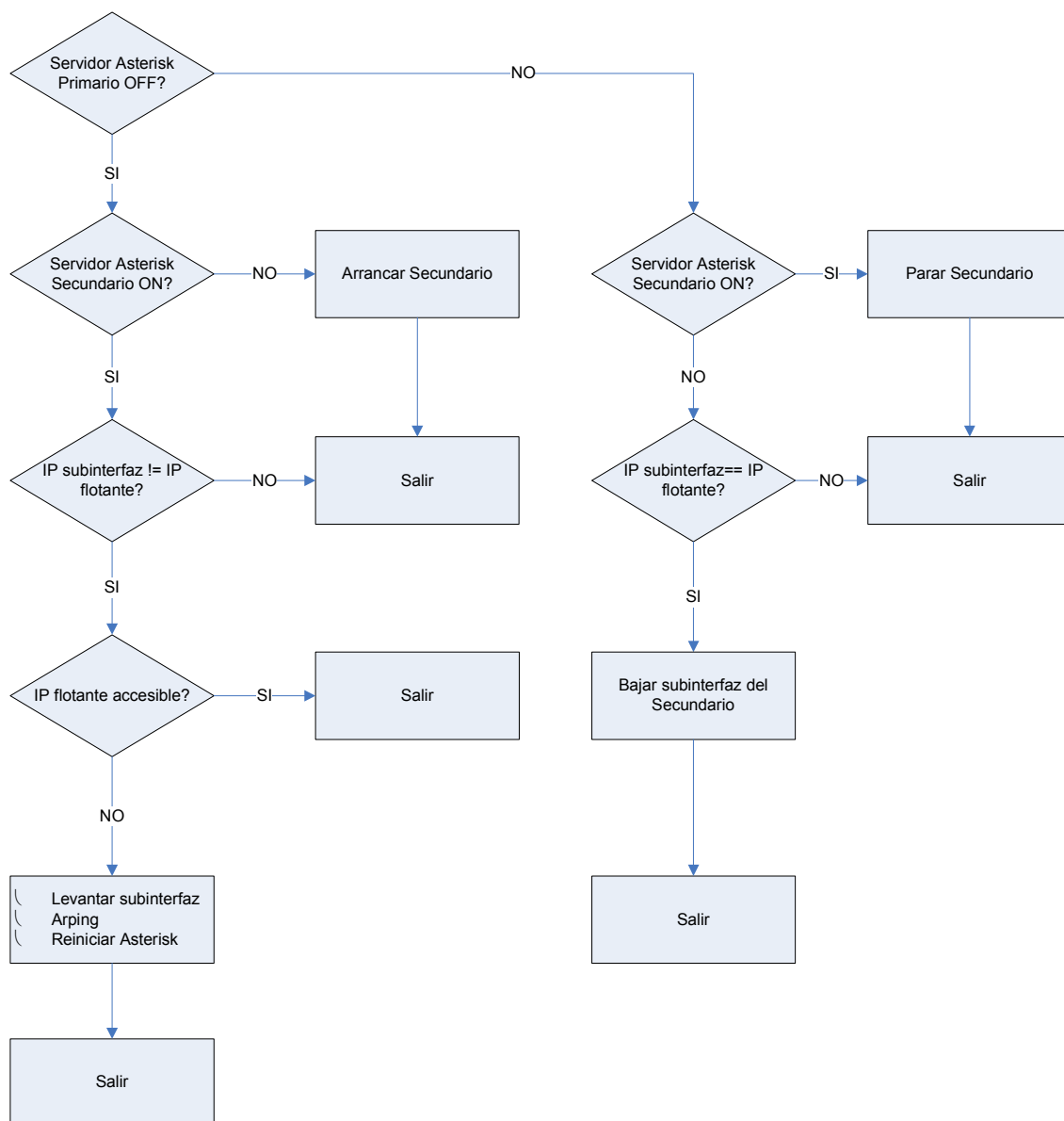


**Ilustración 7.4-2. Esquema Script nodo primario**

La comprobación inicial se realiza sobre el proceso Asterisk (se encuentra en ejecución o no). En caso de que el nodo primario se encuentre en ejecución, se procede a la sincronización con el secundario para mantener la misma configuración en ambos servidores.

En cuanto a su recuperación, destacar el hecho de que si el nodo secundario está caído, el primario no se levantará automáticamente y requerirá de intervención manual. El proceso de restauración consiste en que si el secundario está activo y no existen llamadas realizándose se para el Asterisk secundario, se baja el subinterfaz de red flotante y, posteriormente se arranca Asterisk.

- ESQUEMA SCRIPT NODO SECUNDARIO



**Ilustración 7.4-3. Esquema Script nodo secundario**

Se realiza una comprobación de si el nodo primario está caído, arrancando el servicio secundario si es necesario. En caso de que ambos servicios estén ejecutándose, la primera acción a realizar es parar el servidor secundario para evitar duplicidad.

## 8. ANEXOS

### 8.1 INSTALACIÓN ASTERISK

La instalación de Asterisk se realiza en un ordenador con root local y distribución Linux Fedora 8.

En primer lugar, es necesario descargar el paquete *asterisk-1.6.0.5.tar.gz* de la web de Digium. Posteriormente se descomprime y se instala siguiendo los siguientes pasos:

```
cd /  
mkdir asterisk  
wget http://downloads.digium.com/pub/asterisk/releases/asterisk-1.6.0.5.tar.gz  
tar xzvf asterisk-1.6.0.5.tar.gz  
./configure  
make  
make install  
make samples  
make progdocs
```

También se necesita disponer del paquete *asterisk-addons-1.6.0.1.tar.gz* y del módulo de perl *DBD::mysql*. Los addons añaden ciertas funcionalidades muy interesantes al paquete básico de asterisk, en especial, acceso a bases de datos de mysql y reproducción de audio codificado en mp3. El módulo perl permite el acceso a bases de datos mysql. En primer lugar hay que realizar la instalación del módulo de perl :

```
wget http://search.cpan.org/CPAN/authors/id/C/CA/CAPTTOFU/DBD-mysql-4.013.tar.gz  
tar xzvf DBD-mysql-4.013.tar.gz  
perl Makefile.PL  
make  
make test  
make install
```

Se procede con la instalación del paquete *asterisk-addons*:

```
wget http://downloads.digium.com/pub/asterisk/asterisk-addons-1.6.0.1.tar.gz  
tar xzvf asterisk-addons-1.6.0.1.tar.gz  
cd asterisk-addons  
./configure  
make  
make install
```

Una vez hecho esto, hay que crear el usuario y grupo que se van a encargar de ejecutar Asterisk, se establecen los permisos necesarios para una correcta ejecución.

# Añadimos el grupo y el usuario asterisk

```
/usr/sbin/groupadd asterisk
```

```
/usr/sbin/useradd -d /var/lib/asterisk -g asterisk asterisk
```

# Modificando los propietarios y permisos de los directorios

```
chown --recursive asterisk:asterisk /var/lib/asterisk
chown --recursive asterisk:asterisk /var/log/asterisk
chown --recursive asterisk:asterisk /var/run/asterisk
chown --recursive asterisk:asterisk /var/spool/asterisk
chown --recursive asterisk:asterisk /usr/lib/asterisk
chown --recursive asterisk:asterisk /dev/zap
chown --recursive asterisk:asterisk /dev/dahdi
chmod --recursive u=rwX,g=rX,o= /var/lib/asterisk
chmod --recursive u=rwX,g=rX,o= /var/log/asterisk
chmod --recursive u=rwX,g=rX,o= /var/run/asterisk
chmod --recursive u=rwX,g=rX,o= /var/spool/asterisk
chmod --recursive u=rwX,g=rX,o= /usr/lib/asterisk
chmod --recursive u=rwX,g=rX,o= /dev/dahdi
chown --recursive root:asterisk /etc/asterisk
chmod --recursive u=rwX,g=rX,o= /etc/asterisk
chmod g+w /etc/asterisk/voicemail.conf
chmod g+w,+t /etc/asterisk
```

# Se modifica el script de inicio de asterisk

```
vi /etc/init.d/asterisk
```

```
AST_USER="asterisk"
```

```
AST_GROUP="asterisk"
```

Se comprueba que arranca correctamente y con el usuario establecido.

```
/etc/init.d/asterisk start
```

```
ps -ef | grep asterisk
```

```
root    31297    1  0 /bin/sh /usr/sbin/safe_asterisk -U asterisk -G asterisk
asterisk 31303 31297  0 /usr/sbin/asterisk -f -U asterisk -G asterisk -vvvg -c
```

Por ultimo, se conecta a la consola de Asterisk:

```
asterisk -rvvvv
```

```
Connected to Asterisk 1.6.0.5 currently running on pfc06 (pid = 31303)
```

```
Verbosity is at least 3
```

```
pfc06*CLI>
```

## 8.2 INSTALACIÓN POSTFIX

La instalación de Postfix es necesaria para el envío de los mensajes de voz a direcciones de correo electrónico (voicemail). Los siguientes pasos son una guía de su instalación actual (Los usuarios y contraseñas no se indican por motivos de seguridad):

- Descargar y descomprimir postfix  
postfix-2.5.6.tar.gz
  - Descarga e instalar Cyrus-sasl  
[ftp://ftp.andrew.cmu.edu/pub/cyrus-sasl-2.1.22.tar.gz](ftp://ftp.andrew.cmu.edu/pub/cyrus-mail/cyrus-sasl-2.1.22.tar.gz)  
Descomprimir el paquete
    1. Compilar  
./configure  
make  
make install
    2. Linkamos las librerías:  
[root@pfc06 lib]# ln -s /usr/lib/sasl /usr/local/lib/sasl  
[root@pfc06 lib]# ln -s /usr/lib/sasl2 /usr/local/lib/sasl2
  - Instalación postfix  
En el directorio donde has descomprimido postfix, ejecutar:  
make makefiles CCARGS="-DUSE\_SASL\_AUTH -DUSE\_CYRUS\_SASL \\  
-I/usr/local/include/sasl" AUXLIBS="-L/usr/local/lib -lsasl2"  
make  
Guardar los archivos viejos:  
[root@pfc06 postfix-2.5.6]# mv /usr/sbin/sendmail /usr/sbin/sendmail.OFF  
[root@pfc06 postfix-2.5.6]# mv /usr/bin/newaliases /usr/bin/newaliases.OFF  
[root@pfc06 postfix-2.5.6]# mv /usr/bin/mailq /usr/bin/mailq.OFF  
[root@pfc06 postfix-2.5.6]# chmod 755 /usr/sbin/sendmail.OFF  
/usr/bin/newaliases.OFF \  
➤ /usr/bin/mailq.OFF  
  
Creación de Cuentas y grupos:  
[root@pfc06 postfix-2.5.6]# groupadd -g 12345 postfix  
[root@pfc06 postfix-2.5.6]# useradd -u 12345 -g postfix postfix  
[root@pfc06 postfix-2.5.6]# groupadd -g 54321 postdrop  
Editar /etc/passwd:  
postfix:x:12345:12345::/no/where:/no/shell  
Ejecutamos la instalación del postfix:  
make install  
Please specify the prefix for installed file names. Specify this ONLY  
if you are building ready-to-install packages for distribution to other  
machines.  
install\_root: [/]
- Please specify a directory for scratch files while installing Postfix. You  
must have write permission in this directory.  
tempdir: [/root/postfix/postfix-2.5.6]

Please specify the final destination directory for installed Postfix configuration files.

config\_directory: [/etc/postfix]

Please specify the final destination directory for installed Postfix administrative commands. This directory should be in the command search path of administrative users.

command\_directory: [/usr/sbin]

Please specify the final destination directory for installed Postfix daemon programs. This directory should not be in the command search path of any users.

daemon\_directory: [/usr/libexec/postfix]

Please specify the final destination directory for Postfix-writable data files such as caches or random numbers. This directory should not be shared with non-Postfix software.

data\_directory: [/var/lib/postfix]

Please specify the destination directory for the Postfix HTML files. Specify "no" if you do not want to install these files.

html\_directory: [no]

Please specify the owner of the Postfix queue. Specify an account with numerical user ID and group ID values that are not used by any other accounts on the system.

mail\_owner: [postfix]

Please specify the final destination pathname for the installed Postfix mailq command. This is the Sendmail-compatible mail queue listing command.

mailq\_path: [/usr/bin/mailq] /usr/bin/postfix/mailq

Please specify the destination directory for the Postfix on-line manual pages. You can no longer specify "no" here.

manpage\_directory: [/usr/local/man]

Please specify the final destination pathname for the installed Postfix newaliases command. This is the Sendmail-compatible command to build alias databases for the Postfix local delivery agent.

newaliases\_path: [/usr/bin/newaliases]

Please specify the final destination directory for Postfix queues.

queue\_directory: [/var/spool/postfix]

Please specify the destination directory for the Postfix README files. Specify "no" if you do not want to install these files.

readme\_directory: [no]

Please specify the final destination pathname for the installed Postfix

sendmail command. This is the Sendmail-compatible mail posting interface.  
 sendmail\_path: [/usr/sbin/sendmail] /usr/bin/postfix/sendmail

Please specify the group for mail submission and for queue management commands. Specify a group name with a numerical group ID that is not shared with other accounts, not even with the Postfix mail\_owner account. You can no longer specify "no" here.

setgid\_group: [postdrop]

Updating /usr/libexec/postfix/anvil...

- Arranque postfix

*postfix start*

miramos /var/log/maillog y observamos si ha arrancado correctamente:

Feb 27 11:01:05 pfc06 postfix/postfix-script[21729]: starting the Postfix mail system

Feb 27 11:01:05 pfc06 postfix/master[21730]: daemon started -- version 2.5.6, configuration /etc/postfix

*Modificamos /etc/postfix/main.cf*

mynetworks = 10.6.4.0/24, 127.0.0.0/8,

relayhost = smtp.gmail.com

## Añadimos al final

smtp\_sasl\_auth\_enable = yes

smtp\_sasl\_password\_maps = hash:/etc/postfix/smtp\_pass

smtp\_sasl\_security\_options = noanonymous

smtp\_generic\_maps = hash:/etc/postfix/generic

*Editamos el fichero /etc/postfix/smtp\_pass*

[smtp.gmail.com] <usuario>:<contraseña>

Ejecutamos postmap /etc/postfix/smtp\_pass

Editamos /etc/postfix/generic

root@pfc06.localdomain <usuario>@gmail.com

correo@pfc06.localdomain <usuario>@gmail.com

Ejecutamos postmap /etc/postfix/generic

- Scripts de Arranque postfix

Crear script para arranque: vi /etc/init.d/postfix

```
#!/bin/sh
```

```
#
```

```
# /etc/rc.d/rc.postfix
```

```
#
```

```
POSTFIX=/usr/sbin/postfix
```

```
function start_postfix()
```

```
{
```

```
    echo "Starting Postfix..."
```

```
$POSTFIX start
}

function stop_postfix()
{
    echo "Stopping Postfix..."
    $POSTFIX stop
}

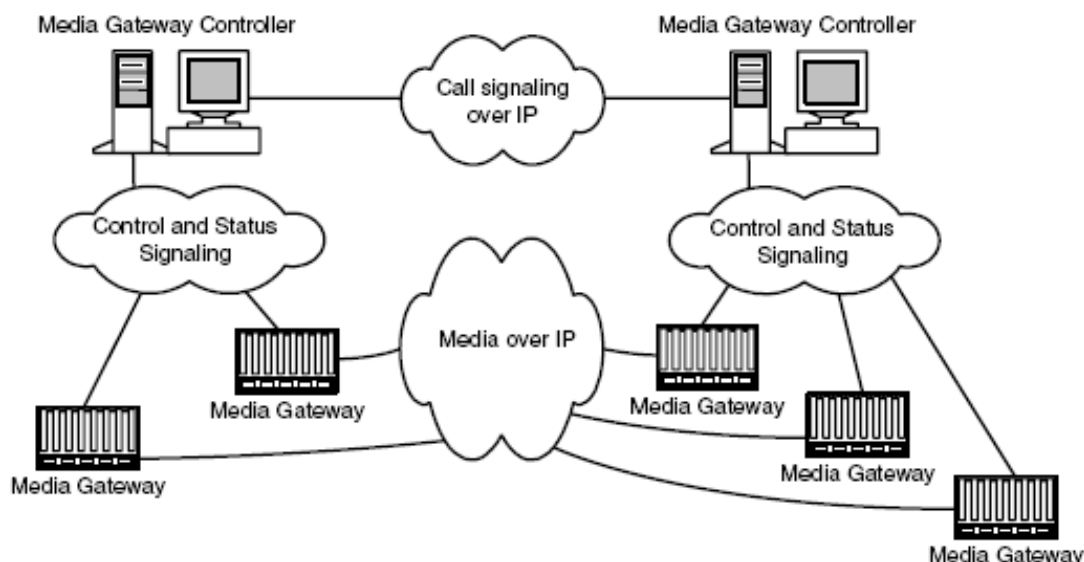
function restart_postfix()
{
    stop_postfix
    sleep 5
    start_postfix
}

case "$1" in
'start')
    start_postfix ;;
'stop')
    stop_postfix ;;
'restart')
    restart_postfix ;;
*)
    echo "usage $0 start|stop|restart" ;;
esac
Añadir postfix a los scripts de arranque (S32Postfix)
```



### 8.3 INTERCONEXIÓN DE MGCs (SIP vs H.323)

Para la interconexión entre *Media Gateway Controllers* es necesario utilizar protocolos como SIP o H.323 para la señalización. Se pretendía haber incluido en la comparativa el protocolo IAX de Asterisk, pero fue desestimado por la falta de información sobre su uso en la interconexión de Media Gateway Controllers.



- Ancho de banda.**  
 H.323 utiliza un menor ancho de banda que SIP ya que los mensajes están codificados de forma binaria mientras que en SIP son texto plano. Además, el texto plano requiere mayor tiempo de procesamiento.
- Fiabilidad**  
 H.323 tiene implementadas funciones para la detección de fallo en dispositivos, siendo una mejora respecto a SIP que detecta este tipo de errores al expirar un temporizador.
- Compatibilidad**  
 La filosofía de H.323 es permanecer compatible con versiones anteriores del protocolo ofreciendo estabilidad y continuidad. SIP no asegura la compatibilidad entre versiones. Concretamente, los RFC 3261 y 2543 no son plenamente compatibles.
- Seguridad**  
 SIP permite la autenticación y cifrado por medio de varios métodos (http, SSL, PGP, etc.). H.323 únicamente lo hace a través de H.234.

## 8.4 INSTALACIÓN DAHDI

La tarjeta integrada de telefonía de la que se dispone es una TDM400P de Digium con dos puertos FXS y dos puertos FXO.

Es necesario la instalación de los drivers para su reconocimiento y utilización:

```
wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/releases/dahdi-
linux-complete-2.2.0+2.2.0.tar.gz
tar -xzyf dahdi-linux-complete-2.2.0+2.2.0.tar.gz
cd dahdi-linux-complete-2.2.0+2.2.0
make
make install
make config
```

Se procede a arrancar el servicio:

```
/etc/init.d/dahdi start
```

Ahora comprobamos si el modulo de Dahdi está cargado en el kernel:

```
[root@pfc06 ~]# dmesg | grep dahdi
dahdi: Telephony Interface Registered on major 196
dahdi: Version: 2.2.0.2
dahdi_transcode: Loaded.
dahdi: Registered tone zone 0 (United States / North America)
dahdi_echocan_mg2: Registered echo canceler 'MG2'
dahdi: Registered tone zone 6 (Spain)
```

Comprobación de correcto funcionamiento:

```
[root@pfc06 ~]# dahdi_cfg -vv
DAHDI Tools Version - 2.2.0

DAHDI Version: 2.2.0.2
Echo Cancellor(s): MG2
Configuration
=====
```

Channel map:

```
Channel 01: FXS Kewlstart (Default) (Echo Canceler: mg2) (Slaves: 01)
Channel 02: FXS Kewlstart (Default) (Echo Canceler: mg2) (Slaves: 02)
Channel 03: FXO Kewlstart (Default) (Echo Canceler: mg2) (Slaves: 03)
Channel 04: FXO Kewlstart (Default) (Echo Canceler: mg2) (Slaves: 04)
```

La configuración física de la tarjeta se realiza mediante el fichero de configuración */etc/dahdi/system.conf*

```
# Autogenerated by /usr/sbin/dahdi_genconf on Thu Apr 15 19:29:24 2010
# If you edit this file and execute /usr/sbin/dahdi_genconf again,
# your manual changes will be LOST.
# Dahdi Configuration File
#
# This file is parsed by the Dahdi Configurator, dahdi_cfg
#
# Span 1: WCTDM/4 "Wildcard TDM400P REV E/F Board 5" (MASTER)
fxsks=1
echocanceller=mg2,1
fxsks=2
echocanceller=mg2,2
fxoks=3
echocanceller=mg2,3
fxoks=4
echocanceller=mg2,4

# Global data

loadzone      = es
defaultzone   = es
```

Con esta configuración el sistema ya es capaz de reconocer y comunicarse adecuadamente con la tarjeta IP, por lo que Asterisk ya puede utilizarla para realizar y recibir llamadas.

## 8.5 INSTALACIÓN MYSQL

Se han instalado dos servidores mysql. El primero para realizar los tests necesarios con la base de datos de alumnos y sus correspondientes calificaciones. El segundo en una máquina virtual, con la finalidad de guardar los logs CDR siendo así plenamente autónoma.

En ambos casos, la instalación del software se ha realizado de forma similar.

Se instala el servidor y cliente de mysql:

```
rpm -U mysql-5.0.45-6.fc8.i386.rpm
rpm -U mysql-devel-5.0.45-6.fc8.i386.rpm
rpm -U mysql-server-5.0.45-6.fc8.i386.rpm
```

Se instala y descarga el DBD:

<http://search.cpan.org/CPAN/authors/id/C/CA/CAPTTOFU/DBD-mysql-4.014.tar.gz>

```
perl -MCPAN -e 'install Bundle::DBD::mysql'
```

## 8.6 ARCHIVOS DE CONFIGURACIÓN

En este apartado reproducen íntegramente los ficheros de configuración:

- **/etc/asterisk/asterisk.conf**

```
[directories] ; remove the (!) to enable this
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astdbdir => /var/lib/asterisk
astkeydir => /var/lib/asterisk
astdatadir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run/asterisk
astlogdir => /var/log/asterisk

;[options]
;verbose = 3
;debug = 3
;alwaysfork = yes ; same as -F at startup
;nofork = yes ; same as -f at startup
;quiet = yes ; same as -q at startup
;timestamp = yes ; same as -T at startup
;execincludes = yes ; support #exec in config files
;console = yes ; Run as console (same as -c at startup)
;highpriority = yes ; Run realtime priority (same as -p at startup)
;initcrypto = yes ; Initialize crypto keys (same as -i at startup)
;nocolor = yes ; Disable console colors
;dontwarn = yes ; Disable some warnings
;dumpcore = yes ; Dump core on crash (same as -g at startup)
;languageprefix = yes ; Use the new sound prefix path syntax
;internal_timing = yes
;systemname = my_system_name ; prefix uniqueid with a system name for global uniqueness issues
;autosystemname = yes ; automatically set systemname to hostname - uses 'localhost' on failure, or systemname
if set
;maxcalls = 10 ; Maximum amount of calls allowed
;maxload = 0.9 ; Asterisk stops accepting new calls if the load average exceed this limit
;maxfiles = 1000 ; Maximum amount of openfiles
;minmemfree = 1 ; in MBs, Asterisk stops accepting new calls if the amount of free memory falls below this
watermark
;cache_record_files = yes ; Cache recorded sound files to another directory during recording
;record_cache_dir = /tmp ; Specify cache directory (used in conjunction with cache_record_files)
;transmit_silence_during_record = yes ; Transmit SLINEAR silence while a channel is being recorded
;transmit_silence = yes ; Transmit SLINEAR silence while a channel is being recorded or DTMF is being
generated
;transcode_via_sln = yes ; Build transcode paths via SLINEAR, instead of directly
;runuser = asterisk ; The user to run as
;rungroup = asterisk ; The group to run as

; Changing the following lines may compromise your security.
;[files]
;astctlpermissions = 0660
;astctlowner = root
;astctlgroup = apache
;astctl = asterisk.ctl

[compat]
pbx_realtime=1.6
res_agi=1.6
app_set=1.6
```

- **/etc/asterisk/cdr\_mysql.conf**

```
;
; Note - if the database server is hosted on the same machine as the
; asterisk server, you can achieve a local Unix socket connection by
; setting hostname=localhost
;
; port and sock are both optional parameters. If hostname is specified
; and is not "localhost" (you can use address 127.0.0.1 instead), then
; cdr_mysql will attempt to connect to the port specified or use the
; default port. If hostname is not specified or if hostname is
; "localhost", then cdr_mysql will attempt to connect to the socket file
; specified by sock or otherwise use the default socket file.
;
[global]
hostname=10.1.1.230
dbname=CDRsFromMGC
table=cdr
password=2202621450
user=mgc
;port=3306
;sock=/tmp/mysql.sock
;
; Older versions of cdr_mysql set the calldate field to whenever the
; record was posted, rather than the start date of the call. This flag
; reverts to the old (incorrect) behavior. Note that you'll also need
; to comment out the "start=calldate" alias, below, to use this.
;compat=no
;
;
; ssl connections (optional)
;ssl_ca=<path to CA cert>
;ssl_cert=<path to cert>
;ssl_key=<path to keyfile>
;
; You may also configure the field names used in the CDR table.
;
```

```
[aliases]
start=calldate
callerid=clid
;src=src
;dst=dst
;dcontext=dcontext
;channel=channel
;dstchannel=dstchannel
;lastapp=lastapp
;lastdata=lastdata
;duration=duration
;billsec=billsec
;disposition=disposition
```

- **/etc/asterisk/chan\_dahdi.conf**

```
[trunkgroups]
; define any trunk groups
[channels]
; hardware channels
; default
disallow = all
allow = ulaw
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
echocancel=yes
echotraining=yes
immediate=no
; define channels
context=phones ; Uses the [internal] context in extensions.conf
signalling=fxs_ks ; Uses FXO signalling for an FXS channel
channel => 3 ; Telephone attached to port 1
channel => 4

context=incoming_calls
signalling=fxs_ks
channel => 2
```

- **/etc/asterisk/dahdi-channels.conf**

```
; Autogenerated by /usr/sbin/dahdi_genconf on Thu Apr 15 19:29:24 2010
; If you edit this file and execute /usr/sbin/dahdi_genconf again,
; your manual changes will be LOST.
; Dahdi Channels Configurations (chan_dahdi.conf)
;
; This is not intended to be a complete chan_dahdi.conf. Rather, it is intended
; to be #include-d by /etc/chan_dahdi.conf that will include the global settings
;

; Span 1: WCTDM/4 "Wildcard TDM400P REV E/F Board 5" (MASTER)
;;; line="1 WCTDM/4/0 FXSKS"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 1
callerid=
group=
context=default

;;; line="2 WCTDM/4/1 FXSKS (In use)"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 2
callerid=
group=
context=default

;;; line="3 WCTDM/4/2 FXOKS (In use)"
signalling=fxo_ks
callerid="Channel 3" <4003>
mailbox=4003
group=5
context=from-internal
channel => 3
callerid=
mailbox=
group=
context=default

;;; line="4 WCTDM/4/3 FXOKS (In use)"
signalling=fxo_ks
callerid="Channel 4" <4004>
mailbox=4004
group=5
context=from-internal
channel => 4
callerid=
mailbox=
group=
context=default
```

- **/etc/asterisk/extensions.conf**

```
[globals]
LINEA=DAHDI/2-1
[general]

[macro-voicemail]
exten => s,1,Dial(${ARG1},20)
exten => s,2,Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER,1,VoiceMail(u${MACRO_EXTEN})
exten => s-NOANSWER,2,Hangup()
exten => s-BUSY,1,VoiceMail(b${MACRO_EXTEN})
exten => s-BUSY,2,Hangup()
exten => _s-.,1,Goto(s-NOANSWER,1)

[default]
exten => s,1,Verbose(1|Unrouted call handler)
exten => s,n,Answer()
exten => s,n,Wait(1)
exten => s,n,Playback(tt-weasels)
exten => s,n,Hangup()

[incoming_calls]
exten => s,1,Answer()
exten => s,2,Echo()

;Los siguientes contextos son para la realizacion de llamadas externas mediante la linea analogica
;Se han separado en tipo de llamadas (provinciales, interprovinciales, a numeros 90X, numeros 80X y
;moviles).
;Para poder realizar estas llamadas, hay que conectar una linea analogica externa en el puerto 2-1 de la
;tarjeta pci instalada en el PC
;que hace de servidor de Asterisk.

[provinciales_analogico]
exten => _9948XXXXXX,1,Dial(${LINEA}/${EXTEN:1})
exten => _9948XXXXXX,2,Congestion()
exten => _9948XXXXXX,102,Congestion()

[interprovinciales_analogico]
exten => _992XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
exten => _992XXXXXXX,2,Congestion()
exten => _992XXXXXXX,102,Congestion()

[90X_analogico]
exten => _990XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
exten => _990XXXXXXX,2,Congestion()
exten => _990XXXXXXX,102,Congestion()

[80X_analogico]
exten => _980XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
exten => _980XXXXXXX,2,Congestion()
exten => _980XXXXXXX,102,Congestion()

[moviles_analogico]
exten => _96XXXXXXX,1,Dial(${LINEA}/${EXTEN:1})
exten => _96XXXXXXX,2,Congestion()
exten => _96XXXXXXX,102,Congestion()

;Los siguientes contextos son para la realizacion llamadas externas mediante la
;maqueta SIP existente en el Departamento de Telematica.
;Se han organizado de manera similar a los contextos analogicos.

[provinciales_sip]
exten => _0948XXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
exten => _0948XXXXXX,2,Hangup()
exten => _0948XXXXXX,102,Hangup()

[interprovinciales_sip]
exten => _092XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
exten => _092XXXXXXX,2,Hangup()
exten => _092XXXXXXX,102,Hangup()

[90X_sip]
exten => _090XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
exten => _090XXXXXXX,2,Hangup()
exten => _090XXXXXXX,102,Hangup()

[80X_sip]
exten => _080XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
exten => _080XXXXXXX,2,Hangup()
exten => _080XXXXXXX,102,Hangup()

[moviles_sip]
exten => _06XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
exten => _06XXXXXXX,2,Hangup()
exten => _06XXXXXXX,102,Hangup()

[todas_sip]
exten => _0XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
exten => _0XXXXXXX,2,Hangup()
exten => _0XXXXXXX,102,Hangup()

[centralita]
exten => 013,1,Dial(${LINEA}/${EXTEN:1})

;El siguiente contexto permite el acceso a una base de datos creada como prueba.
;En ella se espera que el usuario pulse un numero de 4 cifras, correspondiente al codigo de la asignatura
;y un numero de 6 digitos correspondiente al codigo de alumno.
;Con ambos datos hace una consulta a la bbdd mysql y le devolveria el valor de su nota.
```

```
[asignaturas]
exten => s,1,Read(cod_asignatura,en/digits/1,number,4)
exten => s,n,NoOp(${cod_asignatura})
exten => s,n,Read(cod_alumno,en/digits/2,number,6)
exten => s,n,NoOp(${cod_alumno})
exten => s,n,MySQL(Connect connid localhost root asdfghjkl asignaturas)
exten => s,n,MySQL(Query resultid ${connid} SELECT `nota` from `notas_2009` WHERE
`cod_asignatura`=${cod_asignatura} AND `cod_alumno`=${cod_alumno})
exten => s,n,MySQL(Fetch fetchid ${resultid} vari)
exten => s,n,NoOp(${vari})
exten => s,n,MySQL(Disconnect ${connid})
```

;El siguiente contexto corresponde al menu IVR

```
[ivr_menu]
exten => s,1,Background(en/digits/0)
exten => s,n,WaitExten(5)
exten => 1,1,Goto(ivr_menu_1,s,1)
exten => 2,1,Goto(ivr_menu_2,s,1)
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

[ivr_menu_1]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,WaitExten(5)
exten => 1,1,Goto(ivr_menu_1_1,s,1)
exten => 2,1,Goto(ivr_menu_1_2,s,1)
exten => 3,1,Goto(ivr_menu_1_3,s,1)
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

[ivr_menu_1_1]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,Background(en/digits/1)
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})

exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

[ivr_menu_1_2]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,Background(en/digits/2)
exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})

exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

[ivr_menu_1_3]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/1)
exten => s,n,Background(en/digits/3)

exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})

exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

[ivr_menu_2]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Wait(5)
exten => 1,1,Goto(ivr_menu_2_1,s,1)
exten => 2,1,Goto(ivr_menu_2_2,s,1)
exten => 3,1,Goto(ivr_menu_2_3,s,1)
exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

[ivr_menu_2_1]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Background(en/digits/1)

exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})

exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

[ivr_menu_2_2]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Background(en/digits/2)

exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})

exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)
```

```
[ivr_menu_2_3]
exten => s,1,Background(en/digits/0)
exten => s,n,Background(en/digits/2)
exten => s,n,Background(en/digits/3)

exten => s,n,agi(crear_fichero_dinamico.agi,{UNIQUEID})
exten => s,n,Background(en/locucion-${UNIQUEID})
exten => s,n,agi(borrar_fichero_dinamico.agi,{UNIQUEID})

exten => i,1,Goto(s,1)
exten => t,1,Goto(s,1)

;La siguiente extension es la mas larga de todas. En ella van a ir todos los numeros, a nivel interno, que se
usan en la maqueta.
;En primer lugar, el rango de numeracion establecido para esta maqueta es el 5XXX.

[internal]
include => parkedcalls
exten => 1,1,Dial(DAHDI/4-1)
exten => 2,1,Dial(MGCP/end1@MGCP1P1)
exten => 3,1,Dial(MGCP/end2@MGCP1P2)
exten => 3,2,Hangup()
;exten => 4,1,Dial(SIP/SIP2)
exten => 4,1,Dial(SIP/SIP2)
;exten => 5,1,Dial(MGCP/aaln/S0/SU2/0@Ignacio)
exten => 5,1,Dial(MGCP/aaln/S0/SU2/0@Ignacio)
exten => 6,1,Dial(MGCP/aaln/S0/SU2/1@Ignacio)

; NUMERACION CORRESPONDIENTE A LOS ARMARIOS
; Se ha optado porque la numeracion sea 5XXY
; Siendo XX el numero de armario
; Siendo Y 1 o 2, en funcion del puerto al que queramos llamar.

exten => 5011,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO1,rt)
exten => 5012,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO1,rt)

exten => 5021,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO2,rt)
exten => 5022,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO2,rt)

exten => 5031,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO3,rt)
exten => 5032,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO3,rt)

exten => 5041,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO4,rt)
exten => 5042,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO4,rt)

exten => 5051,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO5,rt)
exten => 5052,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO5,rt)

exten => 5061,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO6,rt)
exten => 5062,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO6,rt)

exten => 5071,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO7,rt)
exten => 5072,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO7,rt)

exten => 5081,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO8,rt)
exten => 5082,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO8,rt)

exten => 5091,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO9,rt)
exten => 5092,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO9,rt)

exten => 5101,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO10,rt)
exten => 5102,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO10,rt)

exten => 5111,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO11)
exten => 5112,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO11)
exten => 5113,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO11&MGCP/aaln/S0/SU2/1@ARMARIO11)

exten => 5121,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO12,rt)
exten => 5122,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO12,rt)

exten => 5131,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO13,rt)
exten => 5132,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO13,rt)

exten => 5141,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO14,rt)
exten => 5142,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO14,rt)

exten => 5151,1,Dial(MGCP/aaln/S0/SU2/0@ARMARIO15,rt)
exten => 5152,1,Dial(MGCP/aaln/S0/SU2/1@ARMARIO15,rt)

;Extensiones correspondientes a clientes SIP de Daniel

exten => 5802,1,Dial(SIP/daniel_softphone)
exten => 5802,n,Hangup
exten => 5802,102,Hangup

exten => 5801,1,Dial(SIP/daniel_ip300)
exten => 5801,n,Hangup
exten => 5801,102,Hangup

; Extension 59XX para características avanzadas.
exten => 5901,1,Goto(ivr_menu,s,1) ;IVR

exten => 5902,1,Goto(asignaturas,s,1) ;Acceso a la Base de datos.

exten => 5903,1,Macro(voicemail,SIP/SIP2) ;

exten => 5904,1,agi(wakeup.php) ;Llamada al servicio despertador.
exten => 5904,2,Hangup()
```



```

exten => 5905,1,Meetme(1234) ;Servicio de Conferencia.
exten => 5905,2,Hangup()

;Llamadas al Rango 4XXX se redirigen al servidor SIP 10.6.8.21
exten => _2XXX,1,Dial(SIP/10.1.1.220/${EXTEN})
exten => _2XXX,2,Hangup
exten => _2XXX,102,Hangup

;Llamadas al Rango 4XXX se redirigen al servidor SIP 10.6.8.21
exten => _4XXX,1,Dial(SIP/10.6.8.21/${EXTEN})
exten => _4XXX,2,Hangup
exten => _4XXX,102,Hangup

;Elementos a borrar
exten => 300,1,Transfer(3)

exten => _0XXXXXXX,1,Dial(SIP/10.6.5.250/${EXTEN})
exten => _0XXXXXXX,2,Hangup()
exten => _0XXXXXXX,102,Hangup()

;exten => 5000,1,Dial(SIP/10.6.5.250/0680823398)
exten => 5000,1,NoOp(CANAL = ${CHANNEL})
exten => 5000,n,Dial(SIP/10.6.5.250/0680823398)
exten => 5000,n,Hangup()
exten => 5000,102, Hangup()

exten => 5001,1,Dial(SIP/10.4.88.12/111)

include => asignaturas
[profesores]
include => internal

```

## • /etc/asterisk/features.conf

```

;
; Sample Call Features (parking, transfer, etc) configuration
;

[general]
parkext => 700 ; What extension to dial to park
parkpos => 701-720 ; What extensions to park calls on. These needs to be
context => parkedcalls ; numeric, as Asterisk starts from the start position
; and increments with one for the next parked call.
;parkinghints = no ; Add hints priorities automatically for parking slots (default is no).
;parkingtime => 45 ; Number of seconds a call can be parked for
; (default is 45 seconds)
;comebacktoorigin = yes ; Whether to return to the original calling extension upon parking
; timeout or to send the call to context 'parkedcalltimeout' at
; extension 's', priority '1' (default is yes).
;courtesytone = beep ; Sound file to play to the parked caller
; when someone dials a parked call
; or the Touch Monitor is activated/deactivated.
;parkedplay = caller ; Who to play the courtesy tone to when picking up a parked call
; one of: parked, caller, both (default is caller)
;parkedcalltransfers = caller ; Enables or disables DTMF based transfers when picking up a parked call.
; one of: callee, caller, both, no (default is no)
;parkedcallreparking = caller ; Enables or disables DTMF based parking when picking up a parked call.
; one of: callee, caller, both, no (default is no)
;adsipark = yes ; if you want ADSI parking announcements
;findslot => next ; Continue to the 'next' free parking space.
; Defaults to 'first' available
;parkedmusicclass=default ; This is the MOH class to use for the parked channel
; as long as the class is not set on the channel directly
; using Set(CHANNEL(musicclass)=whatever) in the dialplan

;transferdiggittimeout => 3 ; Number of seconds to wait between digits when transferring a call
; (default is 3 seconds)
;xfersound = beep ; to indicate an attended transfer is complete
;xferfailsound = beeperr ; to indicate a failed transfer
;pickupexten = *8 ; Configure the pickup extension. (default is *8)
;featurediggittimeout = 500 ; Max time (ms) between digits for
; feature activation (default is 500 ms)
;atxfernoanswertimeout = 15 ; Timeout for answer on attended transfer default is 15 seconds.
;atxferdropcall = no ; If someone does an attended transfer, then hangs up before the transferred
; caller is connected, then by default, the system will try to call back the
; person that did the transfer. If this is set to "yes", the callback will
; not be attempted and the transfer will just fail.
;atxferloopdelay = 10 ; Number of seconds to sleep between retries (if atxferdropcall = no)
;atxfercallbackretries = 2 ; Number of times to attempt to send the call back to the transferer.
; By default, this is 2.

; Note that the DTMF features listed below only work when two channels have answered and are bridged
together.
; They can not be used while the remote party is ringing or in progress. If you require this feature you can
use
; chan_local in combination with Answer to accomplish it.

[featuremap]
blindxfer => 1111 ; Blind transfer (default is #) -- Make sure to set the T and/or t option in
the Dial() or Queue() app call!
;disconnect => *0 ; Disconnect (default is *) -- Make sure to set the H and/or h option in the
Dial() or Queue() app call!
;automon => *1 ; One Touch Record a.k.a. Touch Monitor -- Make sure to set the W and/or w
option in the Dial() or Queue() app call!
;atxfer => *2 ; Attended transfer -- Make sure to set the T and/or t option in the Dial()
or Queue() app call!
;parkcall => #72 ; Park call (one step parking) -- Make sure to set the K and/or k option in the
Dial() app call!

```

```
;automixmon => *3 ; One Touch Record a.k.a. Touch MixMonitor -- Make sure to set the X and/or x
option in the Dial() or Queue() app call!

[applicationmap]
; Note that the DYNAMIC_FEATURES channel variable must be set to use the features
; defined here. The value of DYNAMIC_FEATURES should be the names of the features
; to allow the channel to use separated by '#'. For example:
;
; Set(__DYNAMIC_FEATURES=myfeature1#myfeature2#myfeature3)
;
; (Note: The two leading underscores allow these feature settings to be set on
; on the outbound channels, as well. Otherwise, only the original channel
; will have access to these features.)
;
; The syntax for declaring a dynamic feature is the following:
;
;<FeatureName> => <DTMF_sequence>,<ActivateOn>[<ActivatedBy>],<Application>[,<AppArguments>[,MOH_Class]]
;
; FeatureName -> This is the name of the feature used in when setting the
; DYNAMIC_FEATURES variable to enable usage of this feature.
; DTMF_sequence -> This is the key sequence used to activate this feature.
; ActivateOn -> This is the channel of the call that the application will be executed
; on. Valid values are "self" and "peer". "self" means run the
; application on the same channel that activated the feature. "peer"
; means run the application on the opposite channel from the one that
; has activated the feature.
; ActivatedBy -> This is which channel is allowed to activate this feature. Valid
; values are "caller", "callee", and "both". "both" is the default.
; The "caller" is the channel that executed the Dial application, while
; the "callee" is the channel called by the Dial application.
; Application -> This is the application to execute.
; AppArguments -> These are the arguments to be passed into the application.
; MOH_Class -> This is the music on hold class to play while the idle
; channel waits for the feature to complete. If left blank,
; no music will be played.
;
;
; IMPORTANT NOTE: The applicationmap is not intended to be used for all Asterisk
; applications. When applications are used in extensions.conf, they are executed
; by the PBX core. In this case, these applications are executed outside of the
; PBX core, so it does *not* make sense to use any application which has any
; concept of dialplan flow. Examples of this would be things like Macro, Goto,
; Background, WaitExten, and many more.
;
; Enabling these features means that the PBX needs to stay in the media flow and
; media will not be re-directed if DTMF is sent in the media stream.
;
; Example Usage:
;
;testfeature => #9,peer,Playback,tt-monkeys ;Allow both the caller and callee to play
; ;tt-monkeys to the opposite channel
;
;pauseMonitor => #1,self/callee,Pausemonitor ;Allow the callee to pause monitoring
; ;on their channel
;unpauseMonitor => #3,self/callee,UnPauseMonitor ;Allow the callee to unpause monitoring
; ;on their channel
;
;*** Define another parking lot
;
; You can set parkinglot with the CHANNEL dialplan function
; or by setting 'parkinglot' directly in the channel configuration file.
;
;[parkinglot_edvina]
;context => edvinapark
;parkpos => 800-850
;findslot => next

; GROUPS
; Groups are groupings of features defined in [applicationmap]
; that can have their own key mappings.
;
; Groups are defined as a configuration section,
; and can be set as part of DYNAMIC_FEATURES in
; the same way that a normal feature can...
; etc:
;
; Set(DYNAMIC_FEATURES=myGroupName);
;
; example:
; [myGroupName] ; defines the group named myGroupName
; testfeature => #9 ; associates testfeature with the group and the keycode #9
; pauseMonitor ; associates pauseMonitor with the group and the keycode
; ; defined in [applicationmap]
```

- **/etc/asterisk/meetme.conf**

```
[rooms]
conf => 1234; Se crea la habitacion 1234 para admitir conferencias. Se le llama mediante Meetme(1234)
```

- **/etc/asterisk/mgcp.conf**

```
; MGCP Configuration for Asterisk
;
[general]
port = 2427
bindaddr = 10.6.4.6
disallow = all
allow = g726
allow = alaw
allow = ulaw
;allow = gsm (NO SE PUEDE PERMITIR PORQUE EL MEDIA GATEWAY DA COMUNICANDO)
```

```

allow = speech
;allow = g726
;allow = g729
allow = g726aal2
;allow = all
tos = af31
tos_audio = ef

[Ignacio]
callwaiting = no
threewaycalling = yes
transfer = yes
host = 10.1.6.205
context = internal
callerid= " MGCP_1 "
line => aaln/S0/SU2/0
callerid= " MGCP_2 1760 "
line => aaln/S0/SU2/1
line => *

[MGCP1P1]
usecallerid=yes
hidecallerid=no
callwaiting=yes
threewaycalling=yes
transfer=yes
host = 10.6.4.10
context = internal
;context = prueba
line => end1

[MGCP1P2]
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.6.4.12
context = internal
line => end2

[ARMARIO1]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.2
context = internal
callerid= " ARM1_1 "
line => aaln/S0/SU2/0
callerid= " ARM1_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO2]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.3
context = internal
callerid= " ARM2_1 "
line => aaln/S0/SU2/0
callerid= " ARM2_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO3]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.4
context = internal
callerid= " ARM3_1 "
line => aaln/S0/SU2/0
callerid= " ARM3_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO4]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.5
context = internal
callerid= " ARM4_1 "
line => aaln/S0/SU2/0
callerid= " ARM4_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO5]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.6
context = internal
callerid= " ARM5_1 "
line => aaln/S0/SU2/0
callerid= " ARM5_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO6]
callwaiting=no
threewaycalling=yes

```

```

transfer=yes
host = 10.4.88.7
context = internal
callerid= " ARM6_1 "
line => aaln/S0/SU2/0
callerid= " ARM6_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO7]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.8
context = internal
callerid= " ARM7_1 "
line => aaln/S0/SU2/0
callerid= " ARM7_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO8]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.9
context = internal
callerid= " ARM8_1 "
line => aaln/S0/SU2/0
callerid= " ARM8_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO9]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.10
context = internal
callerid= " ARM9_1 "
line => aaln/S0/SU2/0
callerid= " ARM9_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO10]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.11
context = internal
callerid= " ARM10_1 "
line => aaln/S0/SU2/0
callerid= " ARM10_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO11]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.12
context = internal
callerid= " ARM11_1 "
line => aaln/S0/SU2/0
callerid= " ARM11_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO12]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.13
context = internal
callerid= " ARM12_1 "
line => aaln/S0/SU2/0
callerid= " ARM12_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO13]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.14
context = internal
callerid= " ARM13_1 "
line => aaln/S0/SU2/0
callerid= " ARM13_2 "
line => aaln/S0/SU2/1
line => *

[ARMARIO14]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.15
context = internal
callerid= " ARM14_1 "
line => aaln/S0/SU2/0
callerid= " ARM14_2 "
line => aaln/S0/SU2/1

```

```
line => *

[ARMARIO15]
callwaiting=no
threewaycalling=yes
transfer=yes
host = 10.4.88.16
context = internal
callerid= " ARM15_1 "
line => aaln/S0/SU2/0
callerid= " ARM15_2 "
line => aaln/S0/SU2/1
line => *
```

- **/etc/asterisk/modules.conf**

```
;
; Asterisk configuration file
;
; Module Loader configuration file
;

[modules]
autoload=yes
;
; Any modules that need to be loaded before the Asterisk core has been
; initialized (just after the logger has been initialized) can be loaded
; using 'preload'. This will frequently be needed if you wish to map all
; module configuration files into Realtime storage, since the Realtime
; driver will need to be loaded before the modules using those configuration
; files are initialized.
;
; An example of loading ODBC support would be:
;preload => res_odbc.so
;preload => res_config_odbc.so
;
; res_phoneprov requires func_strings.so to be loaded:
;preload => func_strings.so
;
; Uncomment the following if you wish to use the Speech Recognition API
;preload => res_speech.so
;
; If you want, load the GTK console right away.
;
;load => pbx_gtkconsole.so
;load => pbx_gtkconsole.so
;
;load => res_musiconhold.so
;
; Load one of: chan_oss, alsa, or console (portaudio).
; By default, load Chan_oss only (automatically).
;
;noload => chan_alsa.so
;noload => chan_oss.so
;noload => chan_console.so

; Cargamos el modulo de cdr_mysql
load => cdr_addon_mysql.so
```

- **/etc/asterisk/sip.conf**

```
[general]
context=internal
srvlookup=yes
canreinvite=yes
disallow=all
allow=ulaw
allow=g729
autocreatepeer=yes

[softphone] ;softphone creado para pruebas
type=friend
secret=softphone
qualify=yes
nat=no
host=dynamic
context=internal
callwaiting=yes
transfer=yes
dtmf=rfc2833
allow=ulaw

[daniel_softphone] ;softphone correspondiente a Daniel
type=friend
secret=daniel
qualify=yes
nat=no
host=dynamic
context=internal
callwaiting=yes
transfer=yes
dtmf=rfc2833
allow=ulaw
allow=g729

[daniel_ip300] ;Telefono IP300 correspondiente a Daniel
type=friend
secret=daniel
qualify=yes
nat=no
host=dynamic
```

```
context=internal
callwaiting=yes
transfer=yes
dmthf=rfc2833
allow=ulaw
allow=g729
```

- **/etc/asterisk/voicemail.conf**

```
[general]
attach=yes
format=wav
#maxmessage=180
#minmessage=3
#saycid=yes
#maxlogins=3
#maxgreet=60

[default]
;extension => password,nombre_usuario,email_usuario,email_administracion,opciones
5903 => 1234,prueba,correo_usuario@gmail.com,correo_administracion@gmail.com
```

- **/etc/asterisk/zapata.conf**

```
[trunkgroups]

[channels]
usecallerid=yes
hidecallerid=no
callwaiting=yes
transfer=yes
echocancel=yes
echotraining=yes

context=incoming_calls
signalling=fxo_ks
channel => 3

context=internal
signalling=fxs_ls
channel => 1-2
```

- **Script Alta Disponibilidad Nodo Maestro**

```
#!/bin/sh
set -x
#FLIPL405 - Failover Server Solution
#RUN ON PRIMARY ASTERISK SERVER
#ORIGINAL AUTHOR GREGG HANSEN 20080208 &lt;hansen.gregg@gmail.com>;
#MODIFIED BY GREGORY BOEHNLEIN 20090201 &lt;damin@nacs.net>;
#DEPENDENCIES: nmap, arping

# Version 1.0 - 2009-03-25
# - Consolidated Master/Slave scripts into a single script
# - Converted hardcoded interface / IP configuration to variable based
# - Forced Asterisk to issue a "reload" to bind to the floating IP
# - Consolidated external rsync-replicate script to be self-contained

# If set to "1" this is the Master server.
# If commented out, or set to anything else, this will act as if it is a slave
MASTER="1"

# Direccion al script de arranque de asterisk
DIRECTORY="/etc/init.d/asterisk"

# Directorio ejecucion ifconfig
IFCONFIG_DIR="/sbin/ifconfig"

# Directorio ejecucion arping
ARPING_DIR="/sbin/arping"

# The Master and Slave IP Addresses for Replication / Testing
MASTERIP="10.6.4.201"
SLAVEIP="10.6.4.202"

# The IP address that will float between Master and Slave
FLOAT="10.6.4.6"

# The device on which the floating interface exists
DEVICE="eth0"

# The specific interface alias on the device
IFACE="$DEVICE:1"

# Main Code
if [ "$MASTER" == "1" ] ; then
#if Local Asterisk up = 'open|filtered'
STATUS=$(nmap --system-dns -p 4569 -sU 127.0.0.1 | awk '{print $2}' | grep open)
#if primary owns virtual = '207.166.192.51'
PRIMARYIP=$(("$IFCONFIG_DIR" "$IFACE" | grep "$FLOAT" | awk '{print $2}' | sed 's/addr://g')
#if virtual is not pingable = 'down.'
VIRTUALIP=$(nmap --system-dns -sP "$FLOAT" | grep down | awk '{print $4}'))

# Configuration File Replication from Master to Slave
rsync -avzr --rsh=ssh /etc/asterisk/ root@$SLAVEIP:/etc/asterisk/
rsync -avzr --rsh=ssh /var/spool/asterisk/voicemail root@$SLAVEIP:/var/spool/asterisk
rsync -avzr --rsh=ssh /var/lib/asterisk/moh root@$SLAVEIP:/var/lib/asterisk/moh
rsync -avzr --rsh=ssh /var/lib/asterisk/sounds root@$SLAVEIP:/var/lib/asterisk
rsync -avzr --rsh=ssh /usr/src root@$SLAVEIP:/usr/src
```

```
#!/root/rsync_replicate &gt; /dev/null 2&gt; /dev/null
if [ "$STATUS" == "open|filtered" ] ; then ###is primary asterisk up?
if [ "$PRIMARYIP" != "$FLOAT" ] ; then ###does primary not own virtual ip?
if [ "$VIRTUALIP" == "down." ] ; then ###is the virtual IP not pingable?
$IFCONFIG_DIR $IFACE $FLOAT/24 up
arping -U -c 5 -I $DEVICE $FLOAT ###Gratuitous ARP request
$DIRECTORY reload
fi
fi
else ###No esta corriendo asterisk en el maestro.
SECSTATUS=$(nmap --system-dns -p 4569 -sU $SLAVEIP | awk '{print $2}' | grep open)
if [ "$SECSTATUS" == "open|filtered" ] ; then ###is secondary asterisk up?
LLAMADAS=`ssh root@10.6.4.202 'asterisk -rx "core show calls"|grep active|awk '{printf $1}'`
if [ "$LLAMADAS" != "0" ] ; then
exit 1
else
ssh root@10.6.4.202 '/etc/init.d/asterisk stop;ifconfig eth0:1 down'
sleep 5;
VIRTUALIP=$(nmap --system-dns -sP "$FLOAT" | grep down | awk '{print $4}')
echo $VIRTUALIP
if [ "$VIRTUALIP" == "down." ] ; then ###is the virtual IP not pingable?
$IFCONFIG_DIR $IFACE $FLOAT/24 up
arping -U -c 5 -I $DEVICE $FLOAT ###Gratuitous ARP request
$DIRECTORY start
else
$DIRECTORY start
$IFCONFIG_DIR $IFACE down
fi
fi
fi
else # We must be running as Slave node
###if Primary Asterisk up = 'open|filtered'
PRISTATUS=$(nmap --system-dns -p 4569 -sU $MASTERIP | awk '{print $2}' | grep open)
###if Secondary Asterisk up = 'open|filtered'
SECSTATUS=$(nmap --system-dns -p 4569 -sU 127.0.0.1 | awk '{print $2}' | grep open)
###if local owns Virtual = '207.166.192.51'
PRIMARYIP=$(($IFCONFIG_DIR "$IFACE" | grep "$FLOAT" | awk '{print $2}' | sed 's/addr://g')
###if Virtual not pingable = 'down.'
VIRTUALIP=$(nmap --system-dns -sP "$FLOAT" | grep down | awk '{print $4}')

if [ "$PRISTATUS" != "open|filtered" ] ; then ###is primary asterisk down?
if [ "$SECSTATUS" == "open|filtered" ] ; then ###is secondary asterisk up?
if [ "$PRIMARYIP" != "$FLOAT" ] ; then ###does secondary not own virtual ip?
if [ "$VIRTUALIP" == "down." ] ; then ###is the virtual IP not pingable?
$IFCONFIG_DIR $IFACE $FLOAT/24 up
arping -U -c 5 -I $DEVICE $FLOAT ###Gratuitous ARP request
$DIRECTORY reload
fi
fi
else
$DIRECTORY start
fi
else
if [ "$SECSTATUS" == "open|filtered" ] ; then ###primary is up, is secondary up? (there can be only one!)
$DIRECTORY stop
else
echo
if [ "$PRIMARYIP" == "$FLOAT" ] ; then
# If the Primary is up but we still own the Virtual IP, shut it down
$IFCONFIG_DIR $IFACE down
fi
fi
fi
fi
```

## • Script Alta Disponibilidad Nodo Maestro

```
#!/bin/sh
set -x
#FLIP1405 - Failover Server Solution
#RUN ON PRIMARY ASTERISK SERVER
#ORIGINAL AUTHOR GREGG HANSEN 20080208 &lt;hansen.gregg@gmail.com>;
#MODIFIED BY GREGORY BOEHNLEIN 20090201 &lt;damin@nacs.net>;
#DEPENDENCIES: nmap, arping

# Version 1.0 - 2009-03-25
# - Consolidated Master/Slave scripts into a single script
# - Converted hardcoded interface / IP configuration to variable based
# - Forced Asterisk to issue a "reload" to bind to the floating IP
# - Consolidated external rsync-replicate script to be self-contained

# If set to "1" this is the Master server..
# If commented out, or set to anything else, this will act as if it is a slave
MASTER="0"

# Direccion al script de arranque de asterisk
DIRECTORY="/etc/init.d/asterisk"

# Directorio ejecucion ifconfig
IFCONFIG_DIR="/sbin/ifconfig"

# Directorio ejecucion arpping
ARPPING_DIR="/sbin/arping"

# The Master and Slave IP Addresses for Replication / Testing
MASTERIP="10.6.4.201"
SLAVEIP="10.6.4.202"

# The IP address that will float between Master and Slave
FLOAT="10.6.4.6"

# The device on which the floating interface exists
```

```

DEVICE="eth0"

# The specific interface alias on the device
IFACE="$DEVICE:1"

# Main Code
if [ "$MASTER" == "1" ] ; then
# if Local Asterisk up = 'open|filtered'
STATUS=$(nmap --system-dns -p 4569 -sU 127.0.0.1 | awk '{print $2}' | grep open)
# if primary owns virtual = '207.166.192.51'
PRIMARYIP=$(("$IFCONFIG_DIR" "$IFACE" | grep "$FLOAT" | awk '{print $2}' | sed 's/addr://g')
# if virtual is not pingable = 'down.'
VIRTUALIP=$(nmap --system-dns -sP "$FLOAT" | grep down | awk '{print $4}'))

# Configuration File Replication from Master to Slave
rsync -avzr --rsh=ssh /etc/asterisk/ root@$SLAVEIP:/etc/asterisk/
rsync -avzr --rsh=ssh /var/spool/asterisk/voicemail root@$SLAVEIP:/var/spool/asterisk
rsync -avzr --rsh=ssh /var/lib/asterisk/moh root@$SLAVEIP:/var/lib/asterisk/moh
rsync -avzr --rsh=ssh /var/lib/asterisk/sounds root@$SLAVEIP:/var/lib/asterisk
rsync -avzr --rsh=ssh /usr/src root@$SLAVEIP:/usr/src

# /root/rsync replicate &gt; /dev/null 2&gt; /dev/null
if [ "$STATUS" == "open|filtered" ] ; then ###is primary asterisk up?
if [ "$PRIMARYIP" != "$FLOAT" ] ; then ###does primary not own virtual ip?
if [ "$VIRTUALIP" == "down." ] ; then ###is the virtual IP not pingable?
$IFCONFIG_DIR $IFACE $FLOAT/24 up
arping -U -c 5 -I $DEVICE $FLOAT ###Gratuitous ARP request
$DIRECTORY reload
fi
fi
else
$DIRECTORY start
$IFCONFIG_DIR $IFACE down
fi
else # We must be running as Slave node
###if Primary Asterisk up = 'open|filtered'
PRISTATUS=$(nmap --system-dns -p 4569 -sU $MASTERIP | awk '{print $2}' | grep open)
###if Secondary Asterisk up = 'open|filtered'
SECSTATUS=$(nmap --system-dns -p 4569 -sU 127.0.0.1 | awk '{print $2}' | grep open)
###if local owns Virtual = '207.166.192.51'
PRIMARYIP=$(("$IFCONFIG_DIR" "$IFACE" | grep "$FLOAT" | awk '{print $2}' | sed 's/addr://g')
###if Virtual not pingable = 'down.'
VIRTUALIP=$(nmap --system-dns -sP "$FLOAT" | grep down | awk '{print $4}'))

if [ "$PRISTATUS" != "open|filtered" ] ; then ###is primary asterisk down?
if [ "$SECSTATUS" == "open|filtered" ] ; then ###is secondary asterisk up?
if [ "$PRIMARYIP" != "$FLOAT" ] ; then ###does secondary not own virtual ip?
if [ "$VIRTUALIP" == "down." ] ; then ###is the virtual IP not pingable?

$IFCONFIG_DIR $IFACE $FLOAT/24 up
arping -U -c 5 -I $DEVICE $FLOAT ###Gratuitous ARP request
$DIRECTORY reload
fi
fi
else
$DIRECTORY start
fi
else
if [ "$SECSTATUS" == "open|filtered" ] ; then ###primary is up, is secondary up? (there can be only one!)
$DIRECTORY stop
else
echo
if [ "$PRIMARYIP" == "$FLOAT" ] ; then
# If the Primary is up but we still own the Virtual IP, shut it down
$IFCONFIG_DIR $IFACE down
fi
fi
fi
fi

```



## 9. CONCLUSIONES

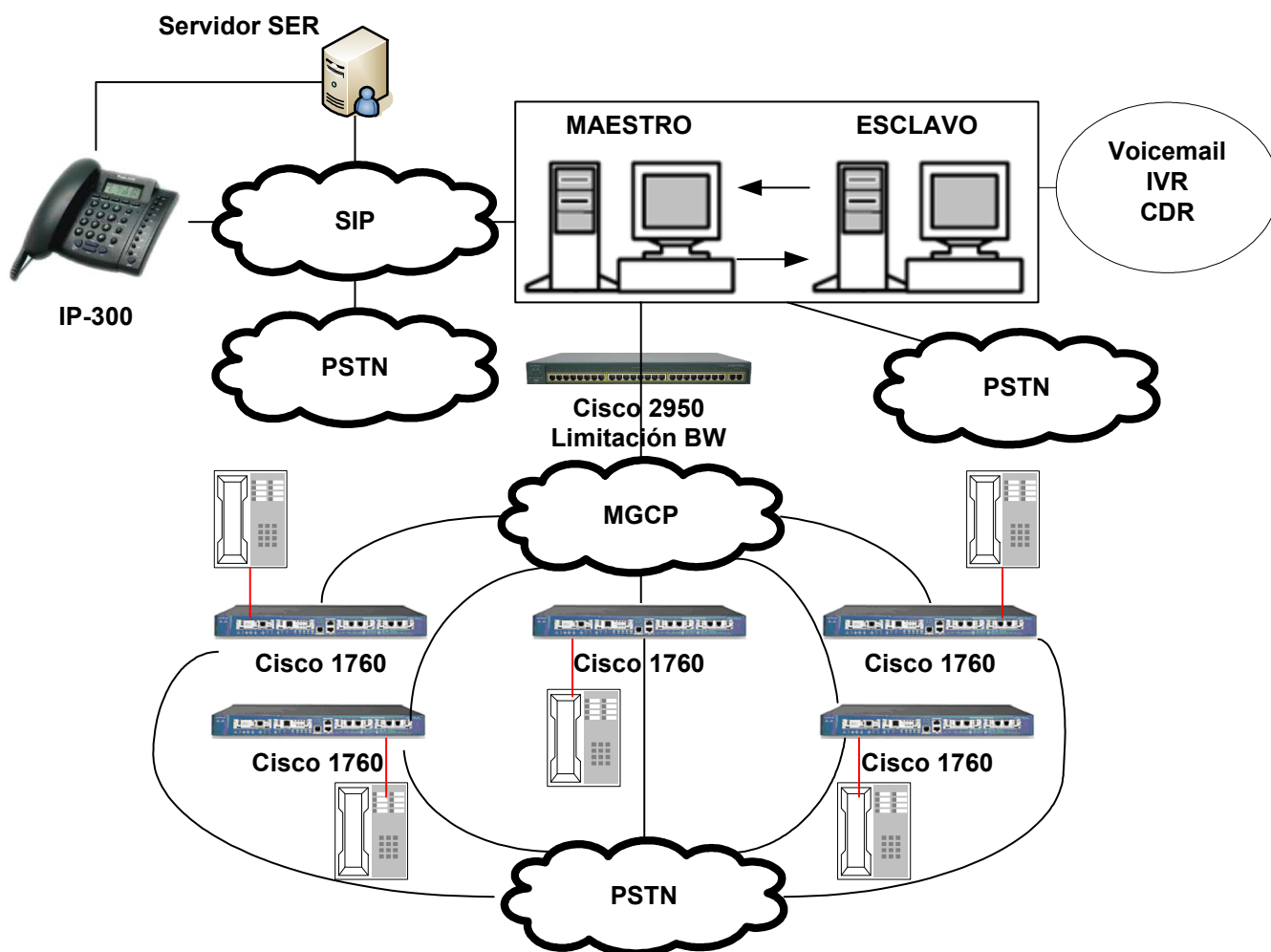
---

Durante el desarrollo del presente proyecto:

- Se ha adquirido un amplio conocimiento acerca de la arquitectura softswitch y del protocolo de control MGCP.
- Se ha instalado un servidor Asterisk que realiza las funciones de Media Gateway Controller, necesario para gestionar una arquitectura softswitch. Se han utilizado routers Cisco 1760 y teléfonos IP-300 como Media Gateways. No obstante, estos últimos presentan deficiencias que hacen totalmente desaconsejable su uso como MGs.
- Se ha establecido un registro de CDRs (Call Detail Records) con el objetivo de efectuar un seguimiento de la utilización de la central telefónica y posibilitar la facturación de las llamadas realizadas.
- Se han configurado los servicios de valor añadido voicemail, IVR, acceso a bases de datos y enrutamiento de llamadas (IP o analógico).
- Se ha integrado esta maqueta con la existente en el Laboratorio de Telemática basada en SIP y con la red PSTN por 3 vías diferentes, permitiendo que usuarios de distintas redes puedan comunicarse entre sí y acceder a todos los servicios.
- Se ha implementado una solución en Alta Disponibilidad permitiendo que, en caso de fallo del servidor primario de Asterisk, un segundo equipo tome el control desempeñando todas las funciones.
- Se ha diseñado una política de QoS que permite 30 llamadas simultáneas con codec G.711 y velocidad de transmisión de 10Mbps/s, independientemente del tráfico de datos. El modelo Catalyst 2950 dispone de 4 colas de prioridad que se han configurado como *weighted round robin*, realizándose 3 estudios (1 teórico y 2 simulaciones prácticas en laboratorio) con el fin de calcular los pesos de cada cola.
  - Estudio Teórico:  
El análisis teórico se ha basado en los parámetros de retardo (inferior a 150ms) y pérdida de paquetes (cola voip estable), obteniéndose pesos 1-1-1-8.
  - Simulación entorno teórico:  
En este escenario se ha simulado un entorno de test lo más parecido posible al caso teórico. En primer lugar, se deshabilitó el uso de ARP, CDP, VTP y *keep-alive*, ya que utilizan paquetes de pequeño tamaño que, con su bajo tiempo de transmisión, reducirían el retardo medio en comparación con el estudio teórico, en el que todos los paquetes de datos son de 1500 bytes. Posteriormente, la generación y análisis de tráfico se realizó mediante el software D-ITG, obteniéndose pesos 1-1-1-9 que cumplen los requerimientos en cuanto a retardo, pérdida de paquetes y jitter (inferior a 10ms). Esta diferencia entre el caso teórico y la simulación se debe, principalmente, al retardo de procesamiento de los paquetes y la utilización de dos switches.
  - Simulación entorno real:  
Se ha estudiado el comportamiento del sistema en un entorno real en el que los protocolos anteriormente deshabilitados estén en funcionamiento, observándose una disminución de la exigencia de los pesos de las colas, siendo suficiente con una distribución 1-1-1-7.

- Se ha establecido un limitador de ancho de banda para evitar que el tráfico clasificado como prioritario supere los 5Mbps/s e interfiera con la red del Departamento de Automática y Computación.

A continuación se muestra un esquema representativo del estado final de la maqueta:



## 10. BIBLIOGRAFÍA

- [1] Daniel Collins. “Carrier Grade Voice Over IP (second edition)”. McGraw-Hill.
- [2] Jim Durkin. “Voice-Enabling the Data Network”. Cisco Systems.
- [3] Ramesh Kaza, Salman Asadullah. “Cisco IP Telephony: Planning, Design, Implementation, Operation, and Optimization”. Cisco Press.
- [4] Jim Van Meggelen, Jared Smith, Leif Madsen. “Asterisk: The Future of Telephony”. O’Reilly Media. 408 pág.
- [5] Cisco Systems. “Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation”.  
[http://conf.com/en/US/tech/tk1077/technologies\\_tech\\_note09186a00800b6710.shtml](http://conf.com/en/US/tech/tk1077/technologies_tech_note09186a00800b6710.shtml)
- [6] Cisco Systems. “Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms)”.  
[http://www.cisco.com/en/US/tech/tk652/tk698/technologies\\_tech\\_note09186a00800945df.shtml](http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a00800945df.shtml)
- [7] Cisco Systems. “Configuring the Cisco IOS MGCP Gateway”.  
[http://www.cisco.com/en/US/products/sw/voicesw/ps556/products\\_tech\\_note09186a008017787b.shtml](http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_tech_note09186a008017787b.shtml)
- [8] Cisco Systems. “Media Gateway Control Protocol (MGCP)”.  
[http://www.cisco.com/en/US/tech/tk652/tk701/tk419/tsd\\_technology\\_support\\_sub-protocol\\_home.html](http://www.cisco.com/en/US/tech/tk652/tk701/tk419/tsd_technology_support_sub-protocol_home.html)
- [9] Cisco Systems. “Sample of Debug MGCP Packets”.  
[http://www.cisco.com/en/US/products/sw/voicesw/ps556/products\\_tech\\_note09186a0080174804.shtml](http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_tech_note09186a0080174804.shtml)
- [10] Cisco Systems. “Verify and Troubleshoot the Cisco IOS MGCP Gateway”.  
[http://www.cisco.com/en/US/products/sw/voicesw/ps556/products\\_tech\\_note09186a008017b2ad.shtml](http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_tech_note09186a008017b2ad.shtml)
- [11] Cisco Systems. “Cisco IOS Voice Command Reference”.  
[http://ciscosystems.com/en/US/docs/ios/12\\_3/vvf\\_r/122tvr.html](http://ciscosystems.com/en/US/docs/ios/12_3/vvf_r/122tvr.html)
- [12] Wikipedia. “Comparison of audio codecs”  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_audio\\_codecs](http://en.wikipedia.org/wiki/Comparison_of_audio_codecs)
- [13] Alessio Botta, Alberto Dainotti, Antonio Pescapè, "Multi-protocol and multi-platform traffic generation and measurement", INFOCOM 2007 DEMO Session, May 2007, Anchorage (Alaska, USA)
- [14] Cisco Systems. “Understanding Delay in Packet Voice Networks”.  
[http://www.cisco.com/en/US/tech/tk652/tk698/technologies\\_white\\_paper09186a00800a8993.shtml](http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800a8993.shtml)